AD-A062 881 TRANSPORTATION SYSTEMS CENTER CAMBRIDGE MASS F/G 1/5
THE AIRPORT PERFORMANCE MODEL. VOLUME II. USER'S MANUAL AND PRO--ETC(U)
OCT 78 J BELLANTONI

UNCLASSIFIED TSC-FAA-28-21.2 FAA-ASP-78-10.2 NL

1 OF 3
ADA
062881

12

Report No. FAA-ASP-78-10,II

Vol-I-AD-A062863

A062863
VOL I

6

THE AIRPORT PERFORMANCE MODEL.
Volume II. User's Manual and Program Documentation.

U.S. Department of Transportation
Research and Special Programs Administration
Transportation Systems Center
Cambridge MA 02142

14 TSC-FAA-28-21.2

18 FAA-ASP 19 78-10.2

12 252p.

11 OCT 1978
FINAL REPORT

9 Final rept.
Jan 75-Jun 76,

DOCUMENT IS AVAILABLE TO THE PUBLIC
THROUGH THE NATIONAL TECHNICAL
INFORMATION SERVICE, SPRINGFIELD,
VIRGINIA 22161

10 J. Bellantoni

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
Federal Aviation Administration
Office of Aviation System Plans
Washington DC 20590

JCB

407 082

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

NOTICE

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

2

Technical Report Documentation Page

1. Report No. FAA-ASP-78-10,II✓	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle THE AIRPORT PERFORMANCE MODEL, Volume II: User's Manual and Program Documentation		5. Report Date October 1978 6. Performing Organization Code
7. Author(s) J. Bellantoni, Editor		8. Performing Organization Report No. DOT-TSC-FAA-78-21,I✓ II ?
9. Performing Organization Name and Address U.S. Department of Transportation Research and Special Programs Administration Transportation Systems Center Cambridge MA 02142		10. Work Unit No. (TRAIS) FP705/R9119 11. Contract or Grant No.
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Office of Aviation System Plans Washington DC 20590		13. Type of Report and Period Covered FINAL REPORT January 1975-June 1976 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract Vol II:

Volume II contains a User's manual and program documentation for the Airport Performance Model. This computer-based model is written in FORTRAN IV for the DEC-10. The user's manual describes the user inputs to the interactive program and gives sample outputs. Program documentation includes a description of the model, its data bases and operating options. The network is described for modifying the data as are the statistical procedures for obtaining annual estimates and 20-year benefits and costs.

17. Key Words Model, computer model, interactive program, airport, investment, benefit, cost	18. Distribution Statement DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 258	22. Price

Reproduction of completed page authorized

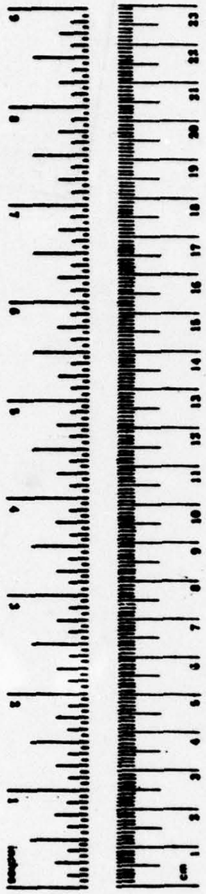
PREFACE

The computer programs described in this volume are the result of work done by Robert Montanari, John F. Dolan, and Ellen Laviana of Kentron International, Inc., Cambridge MA.

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F

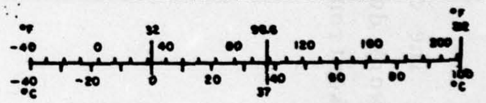


TABLE OF CONTENTS

PART I: COMPOSITE VOLUME TRAFFIC PROFILES FOR APM

Section   Page

1. INTRODUCTION.................................... 1-1

2. OAG PROCESSING.................................. 1-4

2.1 1975 OAG...................................... 1-4
2.2 1972 and 1973 OAG............................. 1-5

3. CLUSTERING...................................... 1-12

3.1 1975 Clustering............................... 1-13
3.2 1972 and 1973 Clustering...................... 1-13

4. WEATHER DATA PROCESSING......................... 1-18

5. TOWER DATA PROCESSING........................... 1-24

6. OAG - TOWER INTERFACE........................... 1-26

7. REVISING THE DATA BASE.......................... 1-28

7.1 OAG... 1-28
7.2 Clustering.................................... 1-34
7.3 Weather....................................... 1-40
7.4 Tower... 1-42
7.5 Annualization................................. 1-45

8. INCIDENTAL PROGRAMS AND DIVERSIONS.............. 1-46

9. PROGRAM DESCRIPTIONS............................ 1-47

9.1 Program OAGRD................................. 1-47
9.2 Program OAGRD2................................ 1-47
9.3 Program OAGPRF................................ 1-48
9.4 Program PRFMRG................................ 1-49
9.5 Program CLUSTR................................ 1-50

9.5.1 Subroutine CLSSUB........................... 1-51
9.5.2 Subroutines EUCLID, HICLUS, PRTREE.......... 1-51
9.5.3 Subroutine SUSPCL........................... 1-51
9.5.4 Subroutine YAWN............................. 1-52

TABLE OF CONTENTS (Cont'd)

Section Page

9.6 Program PRFPLT........................... 1-52

9.6.1 Subroutine PLTSUB.................... 1-52
9.6.2 Subroutine XYPLOT.................... 1-53

9.7 Program CORAL............................ 1-54

9.7.1 Subroutine CCOEFF.................... 1-54
9.7.2 Subroutine SUM....................... 1-54

9.8 Program AVGPRF........................... 1-55

9.9 Program AVGMRG........................... 1-55

9.10 Programs TOWRD and TOWRD3.............. 1-55

9.11 Program TOWMRG.......................... 1-56

9.12 Program TOWER........................... 1-56

9.12.1 Subroutine XFRDIS................... 1-57
9.12.2 Subroutine DISLEV................... 1-58
9.12.3 Subroutine Order.................... 1-58

9.13 Program CLSFRQ.......................... 1-59

9.14 Program APMFG........................... 1-59

9.15 Program FPRINT.......................... 1-59

APPENDIXES TO PART I

APPENDIX A
OAG PROFILE PRODUCTION PROGRAMS.............. A-1

APPENDIX B
CLUSTERING PROGRAMS.......................... B-1

APPENDIX C
WEATHER PROGRAMS............................. C-1

APPENDIX D
TOWER PROGRAMS............................... D-1

APPENDIX E
ANNUALIZATION PROGRAMS....................... E-1

APPENDIX F
INCIDENTAL PROGRAMS.......................... F-1

TABLE OF CONTENTS (Cont'd)

PART II: USER'S MANUAL FOR APM.SAV

Section ... Page

1. INTRODUCTION ... 2-1

2. LOGIN ... 2-3

3. RUNNING THE PROGRAM (AFTER LOGIN) ... 2-5

4. RUNWAY DELAY MODULE ... 2-23

5. SUMMARY OUTPUT ... 2-28

6. GROUNDSIDE SERVICE MODULE ... 2-32

7. LOGGING OFF THE SYSTEM ... 2-33

8. SAMPLE OUTPUT ... 2-34

PART III: APM SYSTEM REPORT

1. INTRODUCTION ... 3-1

2. DESCRIPTION OF THE AIRPORT PERFORMANCE MODEL .. 3-2

2.1 Input to the APM ... 3-2
2.2 Model Output ... 3-2

3. SUBROUTINES ... 3-3

3.1 APM ... 3-3
3.2 PROCS ... 3-3
3.3 ANULY ... 3-4
3.4 HJE1 ... 3-4
3.5 PTOC ... 3-4
3.6 MIXNJ ... 3-5
3.7 PANCAP ... 3-5
3.8 PARAM ... 3-5
3.9 QDLY ... 3-6
3.10 AIROUT ... 3-6

4. COST MODULE ... 3-8

4.1 Description of the Cost Module ... 3-8
4.2 Assumptions of the Cost Module (Air Delay) 3-8
4.3 Assumptions of the Cost Module (Gate Delays ... 3-9
4.4 Input to the Cost Module ... 3-10
4.5 Output of the Cost Module ... 3-10
4.6 DCOST ... 3-11

TABLE OF CONTENTS (Cont'd)

5 POLLUTION MODULE.............................. 3-12

5.1 Description of the Polluciton Module..... 3-12
5.2 Input to the Pollution Module............ 3-12
5.3 Output for the Pollution Module.......... 3-12
5.4 EMIS (POLLUTION)......................... 3-13
5.5 DEPRT (DEPCALC).......................... 3-14

6. GROUNDSIDE PORTION OF THE APM................ 3-15

6.1 Introduction............................. 3-15
6.2 Subroutines.............................. 3-16

6.2.1 GNDCAL............................. 3-16
6.2.2 GNDOUT............................. 3-18

6.3 APM Program Flow Charts.................. 3-19

LIST OF ILLUSTRATIONS

Figure		Page
1	Sample Dendrogram	1-14
2	Sample Output From CORAL	1-15
3	Sample of TDF-14 Data	1-19
4	Sample Weather File	1-22
5	Sample OAGRD2 Batch Control File	1-31
6	Sample OAGRD2 Batch Log File	1-32
7	Sample FOR21.DAT From OAGRD2	1-33
8	Sample Batch Control Files for OAGPRF	1-35
9	Sample Batch Control Files for OAGPRF	1-36
10	Batch Control File for CLUSTR	1-38

LIST OF TABLES

Table		Page
1	AIRPORTS	1-2
2	OAG RECORDS EXTRACTED FOR 31 AIRPORTS	1-6
3	MULTI-LINEAR SORT FOR MAY 1972	1-8
4	ASSIGNMENT OF EQUIPMENT CODES TO AIRCRAFT CLASSES	1-9
5	NAMING CONVENTION FOR QUARTER-HOUR PROFILE FILES FOR 1972 AND 1973	1-10
6	NAMING CONVENTION FOR FILES OF CLUSTERED PROFILES	1-17
7	WEATHER STATION NUMBERS AND CORRESPONDING AIRPORT CODES	1-21
8	STRUCTURES OF FILES FOR INPUT TO APM	1-27
9	TAPES WITH SOURCE DATA (ASCII)	1-29

PART I: COMPOSITE VOLUME TRAFFIC PROFILES FOR APM

1. INTRODUCTION

The aim of this project was the production of composite daily traffic volume profiles for thirty-one airports. These airports are listed in Table 1. The scheduled profiles produced for input to the APM are normalized minute-by-minute profiles, averaged for all days in a given cluster. Three clusters are always used, although capability for variable suspension of clustering has been introduced. Scheduled profiles are generated from extracted and sorted versions of OAG tapes. Weather data is used to separate days for 1972 and 1973 into VFR and IFR days, and subsequent calculations are carried out separately for these two sets of days. Volume distributions for scheduled and non-scheduled operations are obtained from tower data, and these distributions are interfaced with output from the clustering process to produce cluster frequencies. Cluster frequencies are the key to the annualization process, and are used to interface unambiguously tower data for two years (731 days) with OAG data for one month per quarter. The annualization process uses tower, weather, cluster frequency, and clustered scheduled profile files to produce files for input to the APM. These files are of four kinds: daily, header, annualization, and profile. All program source files and save files, as well as data files, produced in the course of this job have been stored on library tape 4368, using TAPE, and can easily be retrieved from it. Program files are also on tape 4747.

Table 1. Airports

BOS	Boston
DCA	Washington National
BAL	Baltimore
EWR	Newark
JFK	Kennedy
LGA	La Guardia
IAH	Houston
PHL	Philadelphia
PIT	Pittsburgh
IAD	Dulles International
FLL	Fort Lauderdale
JAX	Jacksonville
MIA	Miami International
MKE	Milwaukee (Mitchell)
TPA	Tampa International
ATL	Atlanta
CLT	Charlotte, N. C.
ORD	Chicago (O'Hare)
DTW	Detroit (Metro Wayne)
MSP	Minneapolis - St. Paul
CLE	Cleveland
STL	St. Louis (Lambert Field)
MSY	New Orleans
DAL	Dallas (Love Field)

Table 1. (Cont'd)

DEN	Denver
SLC	Salt Lake City
LAX	Los Angeles
SFO	San Francisco
LAS	Las Vegas (McCarren)
SEA	Seattle - Tacoma
HNL	Honolulu International

2. OAG PROCESSING

OAG processing was carried out for two distinct sets of tapes: those for January, February, and March of 1975; and those for February, May, August, and November of 1972 and 1973. Listings of programs referred to in Sections 2.1 and 2.2 comprise Appendix A.

2.1 1975 OAG

OAG data for 1975 was used as test data for the profile and cluster software. Conversion of tapes from EBCDIC to ASCII was uneventful, and extraction (OAGRD) and multi-linear sorting (OAGRD2) was carried out for five airports - ORD, BOS, CLE, SEA, and MIA - without difficulty. Minute-by-minute profiles were produced on tape, and quarter-hour profiles on disk for the five airports. It was discovered that a list of equipment codes used by the profile generation program (OAGPRF) was incomplete, but this did not adversely affect 1975 processing, whose only aim was to compare cluster trees for individual months with those for the quarter. In addition to operations profiles, the profile generation process produced mixes (for five classes), hourly concentrations and arrival/departure ratios, as well as the total volume for each day. The profile generation program is run for one month of OAG data at a time, and requires two tape drives, one for OAG input and one for profile output. The program also uses a temporary disk file of variable size, its upper limit being somewhere below 2000 blocks. The size of this file depends on the number of OAG records being processed. Quarter-hour profiles are stored on disk for future clustering. Both tape and disk output files are unformatted. When testing was being carried out for 1975, disk output for all airports was written to a single file for each month. These were named JAN75.DAT, FEB75.DAT, and MAR75.DAT. Their further adventures are described in Section 3.1.

2.2 1972 and 1973 OAG

OAG tapes for 1972 and 1973 are 7-track BCD with three, or sometimes four, of these tapes containing the records for each month. OAG data for these two years is currently available for one month per quarter. Record length varies from month to month, but is generally fixed within each month. These tapes were converted to ASCII with output for all of the input tapes for a month going to one output tape with three, or sometimes four, files. November 1973 has four files, and May 1973 was expected to have four files. Conversion problems necessitated discounting one of the May files. The number of records lost is assumed to be quite small; even without this file, May 1973 has more records than any other of these eight months. Table 2 records the total number of OAG records extracted for each of the eight months for thirty-one airports.

Table 2. OAG Records Extracted for 31 Airports

Month	No. of Records
February '72	14931
May '72	14857
August '72	15790
November '72	15369
February '73	15476
May '73	16913
August '73	16190
November '73	15131

Extraction and multi-linear sorting were done for thirty-one airports for each month. OAGRD extracts, from the several files on the converted OAG tape, records which are either arrival or departure records for one of the listed airports, and writes a single file on an output tape. OAGRD2 then performs a multi-linear sort of the extracted records, and writes to an output tape separate files for each of the thirty-one airports. This map from extracted records to sorted records is not one-to-one, as any record for which the arrival and departure airports are both on this list, is written to two files on the output tape. This was the case for about one-third of the extracted records. Table 3 contains an example of this process for May 1972.

By the time profile production was begun in earnest, a number of changes had been made in OAGPRF. Tape requirements were the same as for 1975 (See Section 2.1), and a temporary disk file was still being used. Mix classes had been expanded to eight, and a complete listing of equipment codes provided (See Table 4). Any equipment code not listed is assumed to be for a small aircraft (class 7), and in fact class 7 codes are not even listed in the program.

Quarter-hour profiles are still written to disk, but a separate disk file is written for each airport. See Table 5 for the naming convention used for these files. Eight quarter-hour files are produced for each of the 31 airports. Section 3.2

Table 3. Multi-linear Sort for May 1972

Input records = 14857

airport	extracted rec's	airport	extracted rec's
BOS	840	CTL	196
DCA	758	ORD	1995
BAL	323	DTW	593
EWR	609	MSP	431
JFK	1413	CLE	464
LGA	920	STL	587
IAH	458	MSY	387
PHL	728	DAL	852
PIT	674	DEN	541
IAD	213	SLC	193
FLL	233	LAX	1478
JAX	151	SFO	1014
MIA	901	LAS	259
MKE	241	SEA	456
TPA	329	HNL	401
ATL	1182		

Total output records = 19820

Table 4. Assignment of Equipment Codes to Aircraft Classes

<u>Class 1</u> - 4-engine, wide body - 747, 74F

<u>Class 2</u> - 4-engine, standard and stretched - D8S, D8F, B3F, Y62, DC8, 707, SSC, Y76, GAL, T44, STV, V10, SUV, 720, 880, 990, B2F, B7F

<u>Class 3</u> - 3-engine, wide body - D10, L10, A3B

<u>Class 4</u> - 2-engine, standard and stretched - 111, BAC, S11, 737, 73S, VL3, VL6, CVS, CVF, CVL, J60, FAL, DME, DC9, D9S, D9F, FFJ, F28, HAN, HSJ, LJT, T24, T04, T34, VF6, JET, D95, J6

<u>Class 5</u> - 3-engine, standard and stretched - 72S, 727, 72F, CM4, TRD, T1C, T1E, T2E, T3B, Y40, T54

<u>Class 6</u> - large turbo-prop, piston - AVS, AVR, AV, MER, ME, VAN, BRF, C44, CL, CV2, C2, CV3, C3, CV4, C4, CV5, C5, CV6, C6, CVR, A1, A2, A12, A24, A02, AN, A10, A22, A26, ARE, CV, PCU, CU, DBV, B2, PDS, DS, DHC, DHF, DC3, D3, DC4, D4, DC6, D6, D6A, 6A, D6F, D6B, 6B, D6C, DC7, FPR, F27, F7, FH7, FH, PGU, GU, 748, A0, Y14, Y4, Y18, Y8, LEC, LE, LHE, LXA, PHR, HR, MR4, M4, Y51, YS, N26, N2, NAT, T14, T4, VIS, VV, VE, V70, V80, V8, YK2, Y11, SU, A5, CO, N2

<u>Class 7</u> - small aircraft - AL6, A50, A5, ACD, AD, B99, B9, BBR, B8, TB8, TB, B80, BQ, AT1, C45, BDE, B18, B8, TBH, BON, BW, BB0, BTP, BT, BC4, B30, BRF, BRT, BR, BN1, BN, BNT, CAR, CR, CES, CE, PCB, CB, DDV, DO, DDR, HRN, HH, DHO, OT, PR4, R4, PTD, TD, DTO, TO, DTB, D28, DR, C82, GGS, GG, GGM, GM, GW1, GW, GSA, PHA, HA, HLD, HD, PHP, HP, C0V, L4T, L1S, MMB, MVZ, L60, L20, L41, P1A, PP6, PC, PAP, PA, PAZ, PZ, PCH, CH, PNV, PN, PTC, PT, PPS, PS, SKV, SV, PSM, SM, TC4, TC, TPR, PRP, ZZ, 601, 402, TS4, SWM, ST2, MV2

<u>Class 8</u> - other - helicopters and hovercraft - AL0, AL, AB4, AB2, 205, SA3, VT7, HOV, KH4, S34, S61, S1, S55, S58, PS4, S4, 47J, JR

Table 5. Naming Convention for Quarter-hour Profile Files for 1972 and 1973

TLC = a three-letter code

Month	Name of quarter-hour file
February '72	TLC21
May '72	TLC22
August '72	TLC23
November '72	TLC24
February '73	TLC31
May '73	TLC32
August '73	TLC33
November '73	TLC34
merged 8 months	TLCPM

records the rest of their history.

In addition, minute-by-minute profiles now record arrivals and departures separately, and peaking factors are calculated for each day, for later conversion to a peaking factor for the year. For a more detailed discussion of running the profile generation program, see Section 7.1.

3. CLUSTERING

For the purposes of this report, clustering can best be regarded as a process for assigning the elements of a set into smaller subsets with the property that any element of a subset is more similar to the rest of the elements in its subset than it is to the elements in the other subsets. In line with this, the "clusters" here discussed are merely sets of days for a given airport with traffic profiles similar in shape. In all cases, normalized data was clustered, and therefore differences in volume levels do not affect the clustering as used here. Comparisons of clustering for normalized vs. raw data for 1975 showed differences in threshold values but not in cluster assignment.

Existing software for hierarchical clustering was used to analyze OAG traffic profiles for 1975 by producing dendrograms ("trees") to be examined by the initiator. This software is described in detail in report KHL-TSC-76-1396 (Cluster Analysis Programs) dated April 20, 1976. In addition, software was developed for plotting individual daily operations profiles and clustered profiles. Further modifications of the cluster software consisted of the addition of a routine to suspend clustering at three clusters, and a more sophisticated routine to provide, as an option, variable suspension of clustering. Production of dendrograms was not desired for 1972 and 1973.

Programs mentioned in (the following) Sections 3.1 and 3.2 are listed in Appendix B.

3.1 1975 Clustering

Clustering was performed separately for each of the quarter-hour profile files JAN75.DAT, FEB75.DAT, and MAR75.DAT, and dendrograms were produced (program PCLUST). The data on these three files was then merged by PRFMRG, and clustered. Four trees for each of the five airports were given to the initiator, and it was decided that three clustered profiles would be adequate to represent traffic profiles at any airport. Figure 1 provides an example of a dendrogram.

In addition to the trees, a selection of plots of daily profiles and of clustered profiles, produced by program PRFPLT, was provided the initiator. On the basis of the appearance of these plots, it was decided that correlation coefficients would provide a good measure of the validity of using one month of OAG data to represent each quarter. A program (CORAL) was written to correlate, for each of the five airports used in 1975 calculations, three-month clusters with February clusters, February clusters with each other, and days from the three-month period with the February clusters. These correlations, as expected, were quite good. Figure 2 is a sample of the output of the correlation program.

3.2 1972 and 1973 Clustering

For 1972 and 1973, no clustering of individual months was required. PRFMRG merges up to eight monthly quarter-hour profile files into a single file for each of the airports.

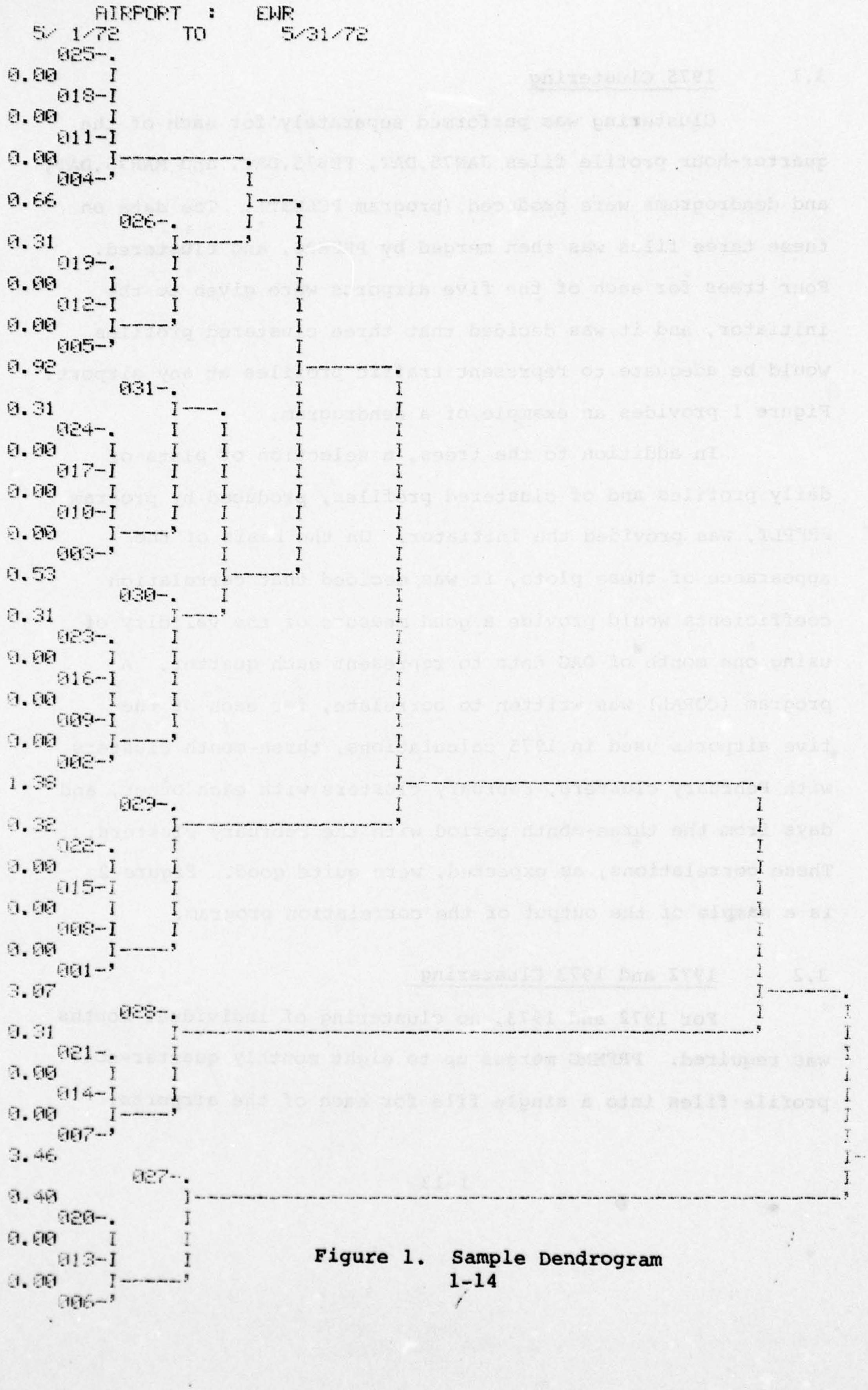
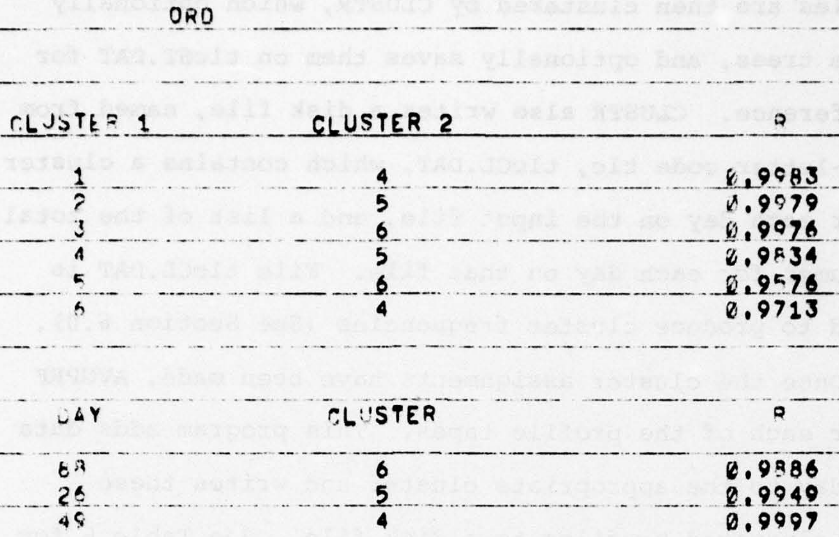


Figure 1. Sample Dendrogram



```
          ORD

CLUSTER 1          CLUSTER 2                R

   1                   4                 Ø.9983
   2                   5                 Ø.9979
   3                   6                 Ø.9976
   4                   5                 Ø.9834
   5                   6                 Ø.9576
   6                   4                 Ø.9713

 DAY               CLUSTER                  R

  68                   6                 Ø.9886
  26                   5                 Ø.9949
  49                   4                 Ø.9997
```

Figure 2. Sample Output From CORAL

The profiles are then clustered by CLUSTR, which optionally prints the trees, and optionally saves them on tlcST.DAT for future reference. CLUSTR also writes a disk file, named from the three-letter code tlc, tlcCL.DAT, which contains a cluster number for each day on the input file, and a list of the total daily volumes for each day on that file. File tlcCL.DAT is later used to produce cluster frequencies (See Section 6.0).

Once the cluster assignments have been made, AVGPRF is run for each of the profile tapes. This program adds data for each day to the appropriate cluster and writes these partially clustered profiles to a disk file. See Table 6 for the naming convention for these files. AVGMRG.F4 then merges them and computes averages for each cluster for all averageable quantities.

When clustering was begun for 1972-73 data, it was found that these clusters did not behave as expected. It was then decided by the initiator that only 1973 OAG data would be used. Files written by AVGPRF were given the last four tags in Table 6, and these four files averaged by AVGMRG.

Table 6. Naming Convention for Files of Clustered Profiles

TLC = three-letter code

Month	File Name
February '72	TLC01
May '72	TLC02
August '72	TLC03
November '72	TLC04
February '73	TLC05
May '73	TLC06
August '73	TLC07
November '73	TLC08
merged file	TLCCP

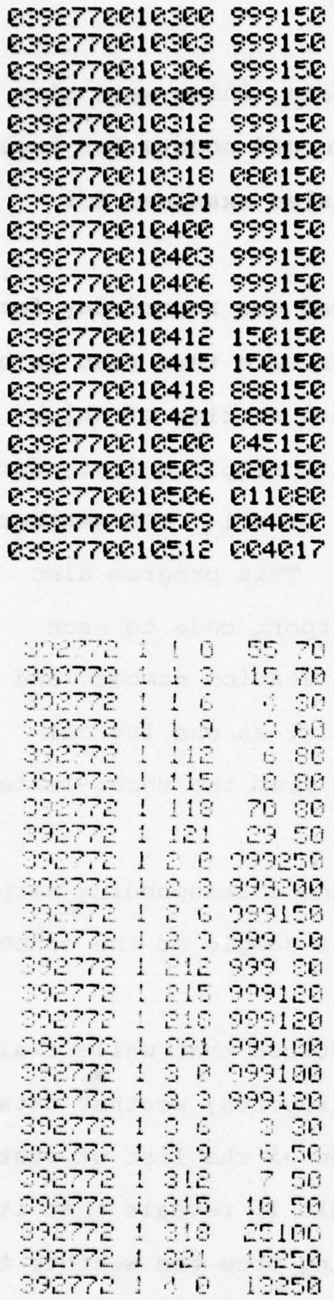
4. WEATHER DATA PROCESSING

A TDF-14 tape containing ceiling and visibility data for 30 stations for the years 1970-1974 was first converted from EBCDIC to ASCII. Figure 3 provides a sample of the data on this tape. Data for 1972 and 1973 were then extracted by EXTWEA and written to tape. Data on this tape consisted of eight observations per day of ceiling and visibility for each day of the two year period. Eight formatted records were thus required for each day at each station, a total of 175,438 records in all (two fewer than expected - see below).

Program WEACOM was then used to compact this ceiling and visibility data, and to write data for each station on a separate file on an output tape. This program also blocked in two missing records for station 24233 (Seattle) for December, 1972.

In general, TDF-14 tapes display a deplorable tendency to be missing records, and sometimes to have extra records. The only information provided regarding these missing records is the months within which they occur, and this only in the form of a monthly record count. A closer examination of the data on these tapes than was deemed necessary for the purposes of this project would be required to establish the precise locations, or rather times, of these missing observations.

Fortunately, the two missing records for Seattle were the only lacunae in the 1972 and 1973 data for the thirty stations used. Numerous other anomalies were observed in



```
039277001030 0 999150
039277001030 3 999150
039277001030 6 999150
039277001030 9 999150
039277001031 2 999150
039277001031 5 999150
039277001031 8 080150
039277001032 1 999150
039277001040 0 999150
039277001040 3 999150
039277001040 6 999150
039277001040 9 999150
039277001041 2 150150
039277001041 5 150150
039277001041 8 888150
039277001042 1 888150
039277001050 0 045150
039277001050 3 020150
039277001050 6 011080
039277001050 9 004050
039277001051 2 004017
```

```
392772 1 1 0  55 70
392772 1 1 3  15 70
392772 1 1 6   4 30
392772 1 1 9   3 60
392772 1 112   6 80
392772 1 115   0 80
392772 1 118  70 80
392772 1 121  29 50
392772 1 2 0 999250
392772 1 2 3 999200
392772 1 2 6 999150
392772 1 2 9 999 60
392772 1 212 999 80
392772 1 215 999120
392772 1 218 999120
392772 1 221 999100
392772 1 3 0 999100
392772 1 3 3 999120
392772 1 3 6   3 30
392772 1 3 9   6 50
392772 1 312   7 50
392772 1 315  10 50
392772 1 318  25100
392772 1 321  15250
392772 1 4 0  13250
```

Figure 3. Sample of TDF-14 Data

record counts for 1970, 1971, and 1974 data, which were also included on the original TDF-14 tape. Record counts for tapes of 10 years of weather data, 1955-1964, also examined for this project, revealed hundreds of gaps.

The compaction procedure produced one record/day for each station, unformatted. These records were then read in by program VFRIFR, which made a list for each station of which days were VFR and which were IFR. This is simply a list, for each station, with 731 entries, each a 0 or a 1. IFR weather is indicated by a 0; VFR weather by a 1. This program also assigned the appropriate three-letter airport code to each station, heretofore identified only by a station number, and wrote the airport code and the VFR/IFR list as one logical record on an unformatted disk file named from the three-letter code, tlcWE.

A list of station numbers and the corresponding airport codes is found in Table 7. Figure 4 is a sample of the content of a weather file.

The reader may, by now, have noticed that while traffic profiles were generated for thirty-one airports, weather data was received for only thirty. Comparison of the list of stations (Table 7) with the list of airports (Table 1) reveals that it is indeed Fort Lauderdale which is missing from the weather tape. Consultation with the initiator further revealed that this was an expected deficiency. Miami's weather was chosen, by reason

Table 7.

Weather Station Numbers and Corresponding Airport Codes

```
STATION NUMBER =    3927     DAL     L.
STATION NUMBER =   12839             MIA
STATION NUMBER =   12842             TPA
STATION NUMBER =   12916             MSY
STATION NUMBER =   12960             IAH
STATION NUMBER =   13739             PHL
STATION NUMBER =   13743             DCA
STATION NUMBER =   13874             ATL
STATION NUMBER =   13881             CLT
STATION NUMBER =   13889             JAX
STATION NUMBER =   13994             STL
STATION NUMBER =   14732             LGA
STATION NUMBER =   14734             EWR
STATION NUMBER =   14739             BOS
STATION NUMBER =   14820             CLE
STATION NUMBER =   14839             MKE
STATION NUMBER =   14922             MSP
STATION NUMBER =   22521             HNL
STATION NUMBER =   23062             DEN
STATION NUMBER =   23169             LAS
STATION NUMBER =   23174             LAX
STATION NUMBER =   23234             SFO
STATION NUMBER =   24127             SLC
STATION NUMBER =   24233             SEA
STATION NUMBER =   93721             BAL
STATION NUMBER =   93738             IAD
STATION NUMBER =   94789             JFK
STATION NUMBER =   94823             PIT
STATION NUMBER =   94846             ORD
STATION NUMBER =   94847             DTW
```

```
           MSP

1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    0    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
0    1    1    1    1    1    1    0    1    1    1    0    1    1    1    1    1    1    1    1
0    1    0    0    1    1    1    1    1    1    1    1    1    1    0    0    1    1    1    1
0    1    1    1    1    0    0    0    1    1    1    1    1    1    1    1    1    1    1    1
1    1    0    0    0    1    1    1    0    1    1    0    0    1    1    1    1    1    1    1
1    0    0    1    1    1    1    1    1    1    1    1    1    0    0    1    1    1    1    1
1    1    1    1    1    1    1    1    1    0    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    0    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    0    1    1    1    1    1    1    0    1    1    1    0    1
1    1    1    1    1    1    1    0    1    1    1    1    1    1    1    0    0    0    0    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    0    0    0    1    1    1
1    1    0    1    1    1    1    1    0    1    1    1    1    1    1    1    0    0    0    1
1    1    1    1    1    1    1    1    1    1    1    1    1    0    0    0    1    1    1    1
1    1    1    0    0    0    0    1    0    0    1    0    0    0    0    0    1    1    1    1
0    1    1    1    0    1    1    1    1    0    1    1    1    1    1    1    1    1    1    0
1    1    1    1    1    1    0    1    0    1    1    1    0    0    0    1    0    0    0    0
1    1    1    0    0    0    1    1    0    1    1    1    1    1    1    1    1    1    1    1
0    1    0    0    0    0    0    0    1    1    1    1    0    1    1    1    0    0    0    0
1    0    1    1    1    1    1    1    1    0    1    1    1    1    0    1    1    1    1    1
1    1    1    0    0    0    0    0    1    1    0    0    1    0    0    0    1    1    0    0
0    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    0    0    1    1
1    1    1    1    0    1    1    1    1    1    0    1    1    1    1    1    1    1    1    1
1    1    1    1    1    0    0    1    1    1    1    0    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    0    1    1    0    0    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    0
0    0    0    0    1    1    1    1    1    1    1    1    1    1    1    1    1    0    1    1
1    1    1    1    0    1    1    1    1    0    1    1    0    1    0    1    1    1    1    1
1    0    1    1    0    0    0    1    1    0    1    1    1    1    1    1    1    1    1    1
1    1    1    1    1    0    1    0    1    1    1    1    1    1    1    1    0    1    1    1
1    1    1    1    1    1    1    1    1    0    0    1    1    0    1    0    1    1    1    1
1    0    0    0    1    1    1    1    1    1    1    1    1    0    1    1    1    1    1    1
1    1    0    0    0    0    0    0    1    1    1
```

Figure 4. Sample Weather File

of spatial disposition, as the best available approximation to Fort Lauderdale's, and a copy of Miami's weather file was made with "FLL" substituted for "MIA".

Listings of the programs discussed in this section are to be found in Appendix C.

5. TOWER DATA PROCESSING

Tapes of FAA form 7230.1 (tower data) were received for 1972 and 1973, four BCD tapes per year. These were converted to ASCII and data for thirty-one airports extracted. This extraction was done by TOWRD for 1972, and TOWRD3 for 1973. The only difference between these two programs is the contents of a list of the locations of the desired airports on the tower data tapes. These lists were different because in 1972, 347 airports filed form 7230.1 with the FAA and in 1973, 353 did. No identification is made on these tapes of airports, and there is no demarcation between data for different airports. Instead, a separate directory is provided on punched cards and listing.

Eight disk files were written by the tower extraction programs. TOWRD wrote TW721.DAT, TW722.DAT, TW723.DAT, and TW724.DAT. TOWRD3 wrote TW731.DAT, TW732.DAT, TW733.DAT, and TW734.DAT. In other words, a separate disk file was produced for each input tape. These files include the three-letter codes for the airports, and are unformatted with one logical record per airport.

These files were merged pairwise by TOWMRG, which takes two input files, one for 1972 and one for 1973, and writes one file for each airport with 1972 and 1973 data merged. These output files are named tlcTW.DAT.

Next, program TOWER separates tower traffic for VFR and IFR days; calculates non-scheduled mixes for each day;

computes scheduled and non-scheduled volume distributions; assigns days to "boxes" on the basis of their scheduled and non-scheduled volumes; calculates average volumes and mixes for each "box"; and computes two-year means for air carrier, air taxi, general aviation, and military traffic.

Input files for TOWER are tlcTW.DAT and tlcWE.DAT. Two output files are produced. File tlcTO.DAT contains two-year means for AC, AT, GA, and MIL traffic, and for non-scheduled mix (one record); for VFR and IFR days, number of scheduled distribution levels, number of non-scheduled distribution levels, number of days in each "box", and averages for scheduled and non-scheduled volumes and non-scheduled mix for each "box", and scheduled and non-scheduled distribution intervals (two logical records - one IFR, one VFR). File tlcIV.DAT contains the number of VFR and IFR days, and scheduled distribution intervals. This file is intended for use by the cluster frequency program (See Section 6.0).

Programs discussed in this section are listed in Appendix D.

6. OAG - TOWER INTERFACE

The key to the interface between OAG data and tower data is provided by CLSFRQ, a program which calculates cluster frequency. Given an upper bound and a lower bound for volumes, a cluster frequency is simply, for the days whose OAG traffic volume falls between the lower bound and the upper bound, the fraction of those days which belong to a given cluster. CLSFRQ reads cluster assignments and daily volumes from tlcCL.DAT, and scheduled volume intervals from tlcIV.DAT and computes cluster fractions (frequencies).

Once the cluster frequencies have been calculated, APMFG can be run. This program reads from files tlcCP.DAT (written by CLUSTR), tlcTO.DAT (written by TOWER), and tlcCF.DAT (written by CLSFRQ), and generates files to be read by the Airport Performance Model (APM).

The output files are tlcDF.DAT (daily file), tlcHF.DAT (header file), tlcAF.DAT (annualization file), and tlcPF.DAT (profile file). Table 8 provides a detailed description of the structures of these files.

Programs discussed in this section are listed in Appendix E.

Table 8. Structures of Files for Input to APM

1.	tlcDF.DAT daily file		record (1)	- $\tilde{V}_S$, $\tilde{V}_N$, $MIX_N(8)$, WTHR(24)
		24X	record (2a)	- Hour (=1,24), $MIX_S^h(8)$, A/D^h, $CONCEN^h$
			record (2b)	- ARR^h, DEP^h
			record (2c)	- P_S^h (60,2)
	WTHR default is 24 x 1 = VFR.			
2.	tlcHF.DAT header file		record (1)	tlc, PF, N_D^V, N_D^I, NRUNS
			record (2)	$\tilde{V}_{AC}^V$, $\tilde{V}_{AT}^V$, $\tilde{V}_{GA}^V$, $\tilde{V}_{MIL}^V$, $\tilde{V}_{AC}^I$, $\tilde{V}_{AT}^I$, $\tilde{V}_{GA}^I$, $\tilde{V}_{MIL}^I$
3.	tlcAF.DAT annualization file		record (1)	NRUNS(box)[≡3), n_1, n_2, n_3
			record (2)	$\tilde{V}_S$(box), $\tilde{V}_N$(box), MIX_N(box)(8), IFR/VFR(0/1)
			record (3a,b,c)	{n_i(run #), p_i(profile #=1,2,3), w_i(box)}, i = 1, 2, 3
4.	tlcPF.DAT profile file		record (1)	profile number = 1, 2, 3
		3X 24X	record (2a)	Hour(=1,24), $MIX_S^{*h}(8)$, A/D^{*h}, $CONCEN^{*h}$
			record 2B)	ARR^{*h}, DEP^{*h}
			record (2c)	P_S^{*h} (60,2)

All files are unformatted. Each line corresponds to one logical record.

7. REVISING THE DATA BASE

Two kinds of changes are possible in the data base generated for the APM: 1) expansion by adding data for additional airports for 1972-1973, 2) use of data for some other year or years. The following sections provide a detailed, though not necessarily complete, description of program modifications and production runs that would be required to make either of these changes. Table 9 lists the source data that was used in preparing APM files for 1972-73.

7.1 OAG

OAG tapes are required for one month of each quarter of a year or years. If the 72-73 data base is being expanded to include other airports, the OAG tapes listed in Table 9 are the ones that will be needed. If another year is being done another set of tapes will be needed.

OAGRD will be run for each OAG tape. If the list of airports is not the same as the list in the program, it will be necessary to make the following changes:

Dimension AN and ANS as the number of airports on the list, and substitute that number for 31 in the statement "IF (NA.NE.31) go to ...". Substitute the list of airports for those in the statement DATA ANS/'BOS', ..., 'HNL'/, make the same changes in OAGRD2, and OAGPRF. If the same list of airports for a different year is being used, no changes are required.

Table 9. Tapes With Source Data (ASCII)

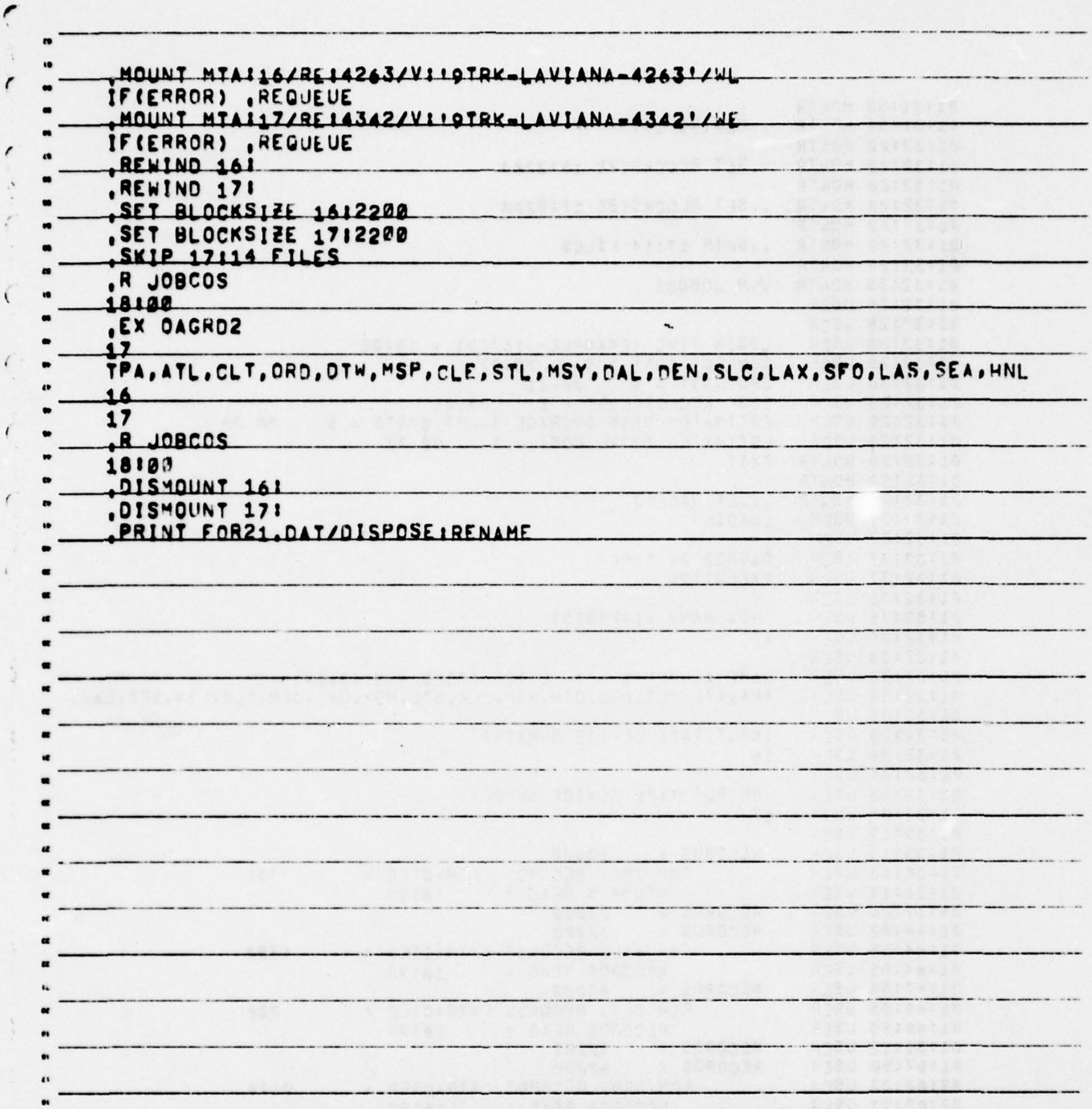
OAG	Month	Reel ID	Blocksize	# Files	Assigned To
	Feb. 1972	4465	2100	3	CONDELL
	May 1972	4466	2200	3	"
	Aug. 1972	4467	2200	3	"
	Nov. 1972	4620	2200	3	"
	Feb. 1973	4370	2200	3	"
	May 1973	3787	2200	3	"
	Aug. 1973	4752	2200	3	"
	Nov. 1973	4762	2200	4	"

Tower	Year	Airports	Reel ID	Blocksize	Assigned To
	1972	1-100	2615	540	CONDELL
	1972	101-200	2622	540	"
	1972	201-300	2623	540	"
	1972	301-347	0035	540	"
	1973	1-100	0075	540	"
	1973	101-200	0167	540	"
	1973	201-300	0202	540	"
	1973	301-353	0460	540	"

Weather	Years	Reel ID	Blocksize	Assigned To
	1970-1974	0468	625	CONDELL
	1972-1973	0467	625	"

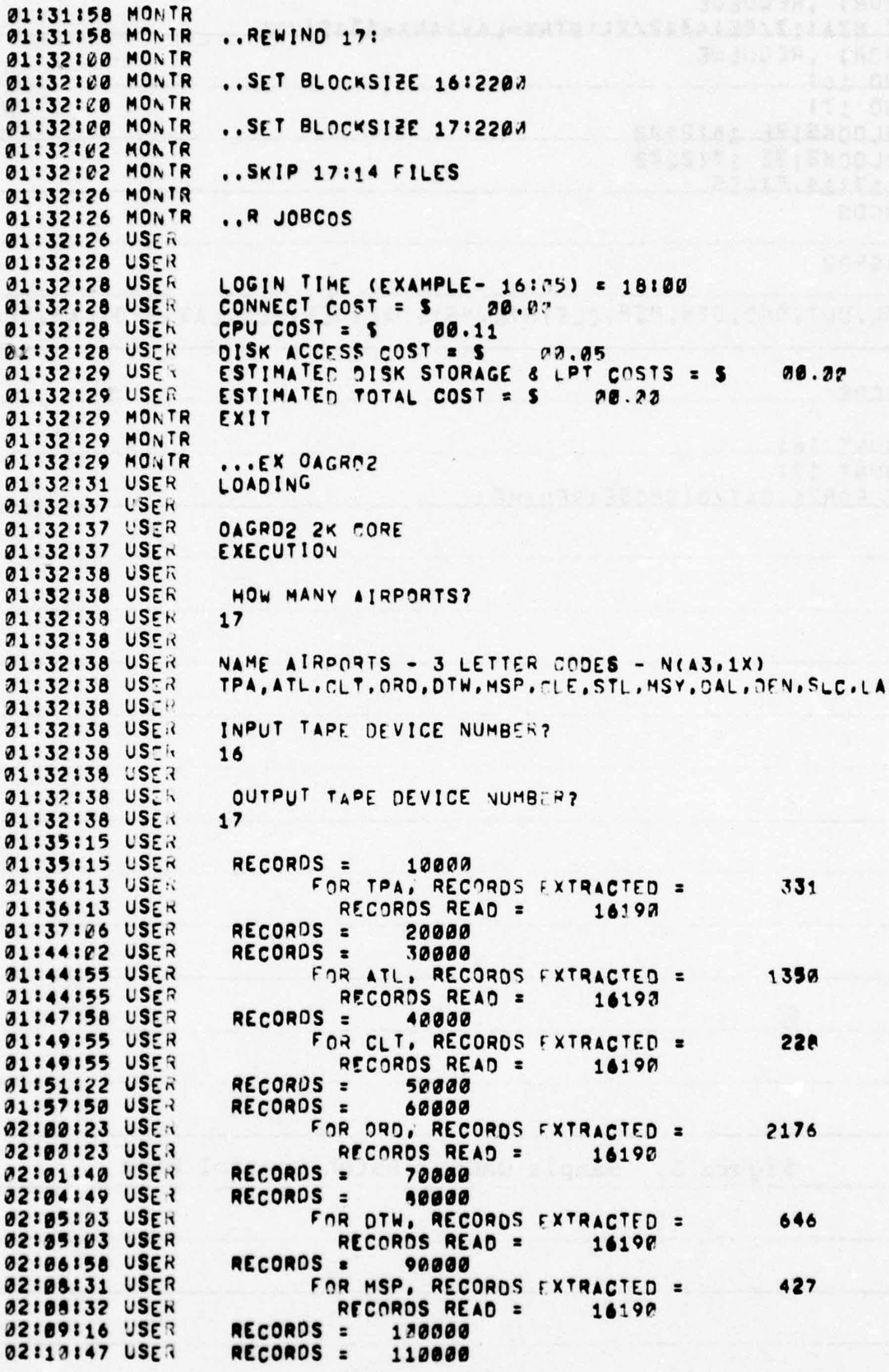
One run of OAGRD will be made for each month of OAG data. This program reads a specified number of files from an input tape and extracts those records whose arrival or departure codes are on the list of airports. An available option is to type in a list of airport codes rather than extract records for all of the airports on the list.

Figure 5 is a sample batch control file for OAGRD2, and Figure 6 part of its log file. The input is similar to that for OAGRD. In this example, fourteen of the airports have already been sorted, and these files are skipped on the output tape. Since fewer than 31 airports are left to be sorted, a list is read in of those remaining. This feature is useful in the event of system crashes, parity errors, hung devices or other misfortunes which may occur during program execution, as only the output for the airport being sorted out at the time is lost. OAGRD2 also produces a disk file called FOR21.DAT, which lists the number of records written for each airport. This file was historically significant, but no longer has a function. An example, produced by the run in Figure 5, is given in Figure 7. The same is true for OAGPRF. Batch control files for this program are shown in Figures 8 and 9. Figure 8 is for a complete run, Figure 9 for a partial run. Note that files are skipped on both input and output devices. If the files on the input tape are not arranged in the same order



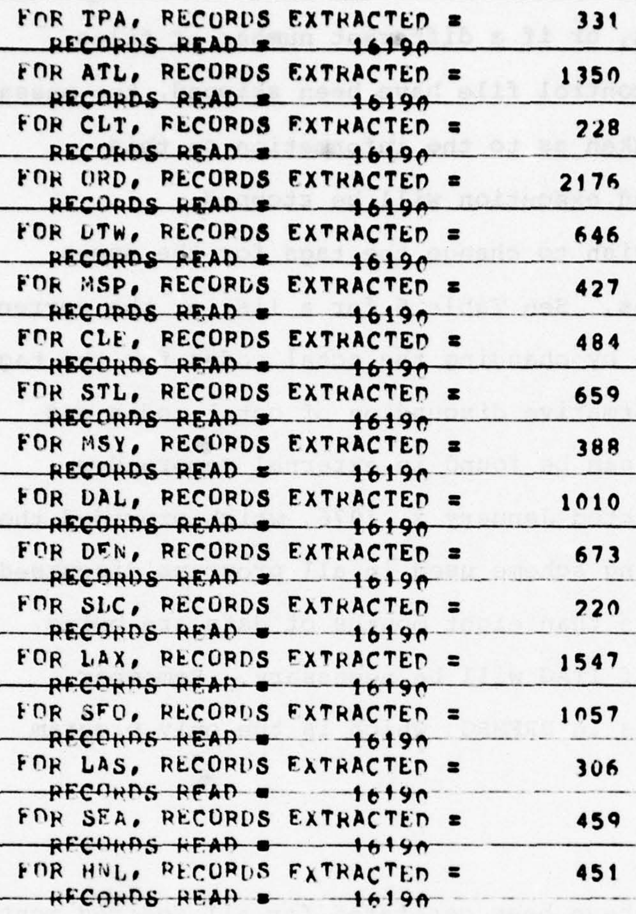
```
,MOUNT MTA:16/RE:4263/V:'9TRK-LAVIANA-4263'/WL
IF(ERROR) ,REQUEUE
,MOUNT MTA:17/RE:4342/V:'9TRK-LAVIANA-4342'/WE
IF(ERROR) ,REQUEUE
,REWIND 16:
,REWIND 17:
,SET BLOCKSIZE 16:2200
,SET BLOCKSIZE 17:2200
,SKIP 17:14 FILES
,R JOBCOS
18:00
,EX OAGRD2
17
TPA,ATL,CLT,ORD,DTW,MSP,CLE,STL,MSY,DAL,DEN,SLC,LAX,SFO,LAS,SEA,HNL
16
17
,R JOBCOS
18:00
,DISMOUNT 16:
,DISMOUNT 17:
,PRINT FOR21.DAT/DISPOSE:RENAME
```

Figure 5. Sample OAGRD2 Batch Control File



```
01:31:58 MONTR
01:31:58 MONTR   ..REWIND 17:
01:32:00 MONTR
01:32:00 MONTR   ..SET BLOCKSIZE 16:2200
01:32:00 MONTR
01:32:00 MONTR   ..SET BLOCKSIZE 17:2200
01:32:02 MONTR
01:32:02 MONTR   ..SKIP 17:14 FILES
01:32:26 MONTR
01:32:26 MONTR   ..R JOBCOS
01:32:26 USER
01:32:28 USER
01:32:28 USER    LOGIN TIME (EXAMPLE- 16:05) = 18:00
01:32:28 USER    CONNECT COST = $    00.02
01:32:28 USER    CPU COST = $    00.11
01:32:28 USER    DISK ACCESS COST = $    00.05
01:32:29 USER    ESTIMATED DISK STORAGE & LPT COSTS = $    00.02
01:32:29 USER    ESTIMATED TOTAL COST = $    00.20
01:32:29 MONTR   EXIT
01:32:29 MONTR
01:32:29 MONTR   ...EX OAGRD2
01:32:31 USER    LOADING
01:32:37 USER
01:32:37 USER    OAGRD2 2K CORE
01:32:37 USER    EXECUTION
01:32:38 USER
01:32:38 USER     HOW MANY AIRPORTS?
01:32:38 USER    17
01:32:38 USER
01:32:38 USER    NAME AIRPORTS - 3 LETTER CODES - N(A3,1X)
01:32:38 USER    TPA,ATL.CLT,ORD,DTW,MSP.CLE,STL,MSY,DAL,DEN,SLC.LAX,SFO.LAS,
01:32:38 USER
01:32:38 USER    INPUT TAPE DEVICE NUMBER?
01:32:38 USER    16
01:32:38 USER
01:32:38 USER     OUTPUT TAPE DEVICE NUMBER?
01:32:38 USER    17
01:35:15 USER
01:35:15 USER     RECORDS =      10000
01:36:13 USER           FOR TPA, RECORDS EXTRACTED =         331
01:36:13 USER             RECORDS READ =       16190
01:37:06 USER     RECORDS =      20000
01:44:02 USER     RECORDS =      30000
01:44:55 USER           FOR ATL, RECORDS EXTRACTED =        1350
01:44:55 USER             RECORDS READ =       16190
01:47:58 USER     RECORDS =      40000
01:49:55 USER           FOR CLT, RECORDS EXTRACTED =         220
01:49:55 USER             RECORDS READ =       16190
01:51:22 USER     RECORDS =      50000
01:57:50 USER     RECORDS =      60000
02:00:23 USER           FOR ORD, RECORDS EXTRACTED =        2176
02:00:23 USER             RECORDS READ =       16190
02:01:40 USER     RECORDS =      70000
02:04:49 USER     RECORDS =      80000
02:05:03 USER           FOR DTW, RECORDS EXTRACTED =         646
02:05:03 USER             RECORDS READ =       16190
02:06:58 USER     RECORDS =      90000
02:08:31 USER           FOR MSP, RECORDS EXTRACTED =         427
02:08:32 USER             RECORDS READ =       16190
02:09:16 USER     RECORDS =     100000
02:10:47 USER     RECORDS =     110000
```

Figure 6. Sample OAGRD2 Batch Log File



```
FOR TPA, RECORDS EXTRACTED =        331
   RECORDS READ =       16190
FOR ATL, RECORDS EXTRACTED =       1350
   RECORDS READ =       16190
FOR CLT, RECORDS EXTRACTED =        228
   RECORDS READ =       16190
FOR ORD, RECORDS EXTRACTED =       2176
   RECORDS READ =       16190
FOR DTW, RECORDS EXTRACTED =        646
   RECORDS READ =       16190
FOR MSP, RECORDS EXTRACTED =        427
   RECORDS READ =       16190
FOR CLE, RECORDS EXTRACTED =        484
   RECORDS READ =       16190
FOR STL, RECORDS EXTRACTED =        659
   RECORDS READ =       16190
FOR MSY, RECORDS EXTRACTED =        388
   RECORDS READ =       16190
FOR DAL, RECORDS EXTRACTED =       1010
   RECORDS READ =       16190
FOR DEN, RECORDS EXTRACTED =        673
   RECORDS READ =       16190
FOR SLC, RECORDS EXTRACTED =        220
   RECORDS READ =       16190
FOR LAX, RECORDS EXTRACTED =       1547
   RECORDS READ =       16190
FOR SFO, RECORDS EXTRACTED =       1057
   RECORDS READ =       16190
FOR LAS, RECORDS EXTRACTED =        306
   RECORDS READ =       16190
FOR SEA, RECORDS EXTRACTED =        459
   RECORDS READ =       16190
FOR HNL, RECORDS EXTRACTED =        451
   RECORDS READ =       16190
```

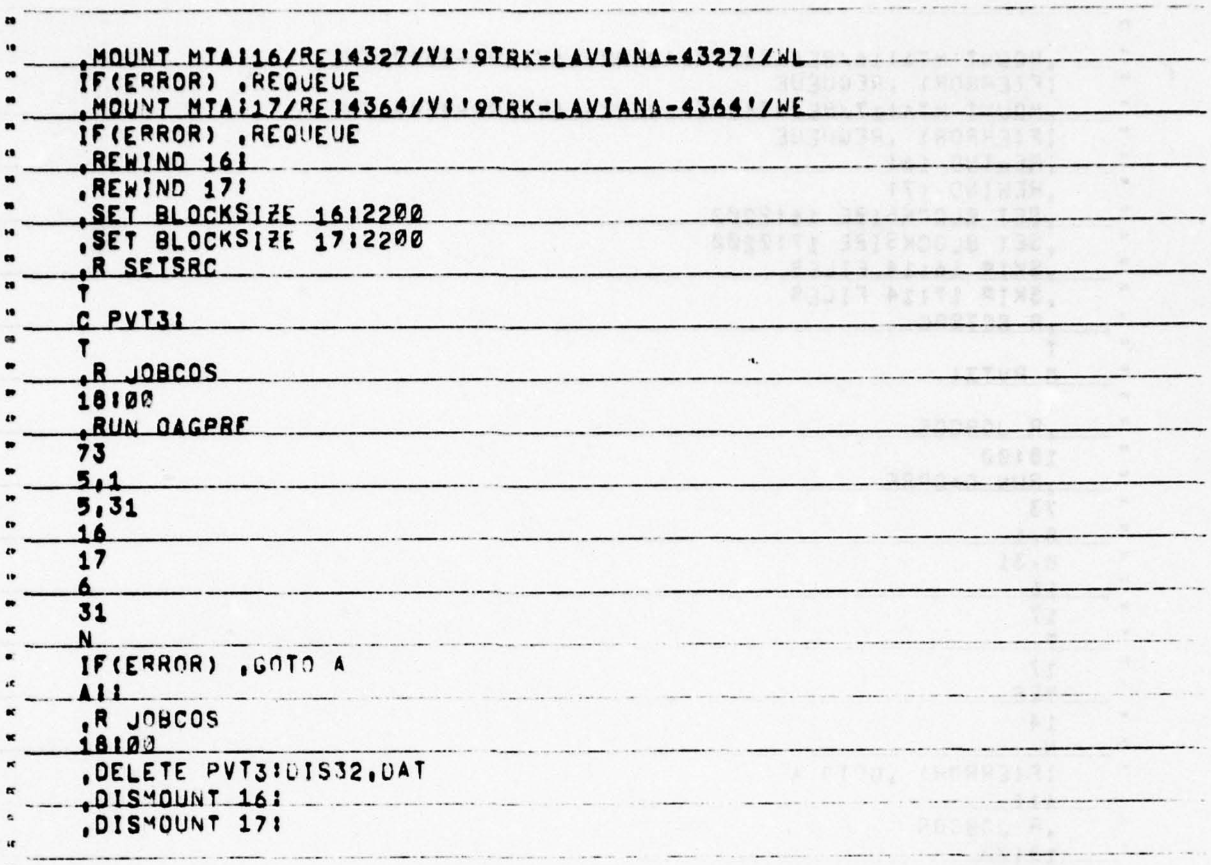
Figure 7. Sample FOR21.DAT From OAGRD2

as the list of airports, or if a different number of files than indicated in the control file have been skipped, tne message "you are somewhat mistaken as to the information on this tape" will be given, and execution will be stopped.

The user may wish to change the tags for the names of the output disk files. See Table 5 for a list of the current tags. This can be done by changing the octal codes for the tags in array ITAG. An informative discussion of octal codes and character manipulation can be found in external memorandum KH10-F29200-EM101/76, dated January 2, 1976, which provided the basis for the file-naming scheme used in all programs discussed in this report. If more than eight months of data are being used, re-dimensioning of ITAG will be necessary. Remember to make the same changes in PRFMRG, which is the only program which uses these files.

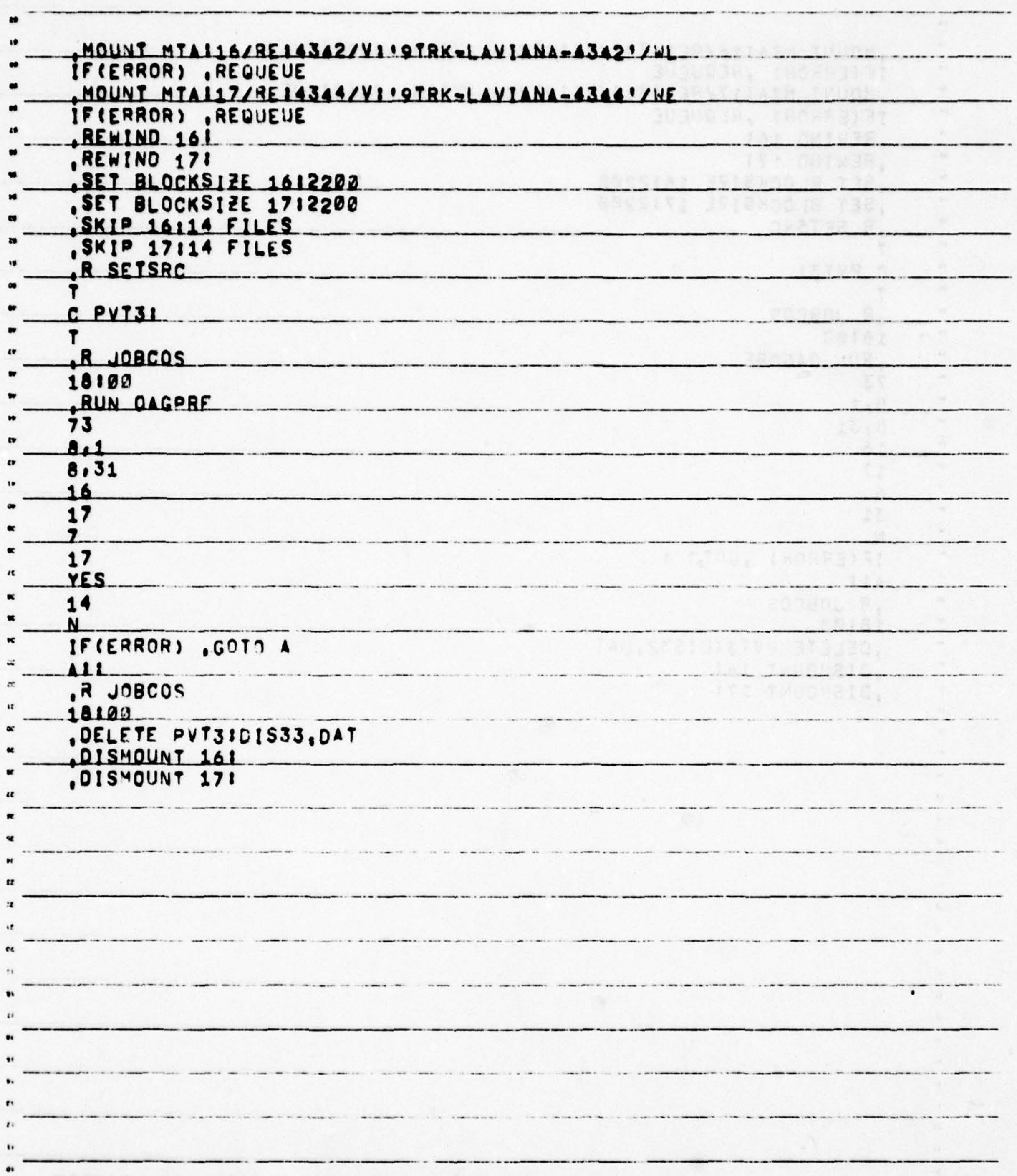
7.2 Clustering

Once profiles have been generated for all desired months, they can be clustered. Clustering cannot be done piecemeal; rather, it is done for all the days at once. Preparatory to clustering, PRFMRG is run. If there is a revised list of airports insert it here. Part of the terminal input is the number of the first and last input files to be used. Usually, these numbers are 1 and the number of months. The program looks for input files with tags corresponding to the file numbers;



```
,MOUNT MTA:16/RE:4327/V:'9TRK-LAVIANA-4327'/WL
IF(ERROR) ,REQUEUE
,MOUNT MTA:17/RE:4364/V:'9TRK-LAVIANA-4364'/WE
IF(ERROR) ,REQUEUE
,REWIND 16:
,REWIND 17:
,SET BLOCKSIZE 16:2200
,SET BLOCKSIZE 17:2200
,R SETSRC
T
C PVT3:
T
,R JOBCOS
18:00
,RUN OAGPRF
73
5,1
5,31
16
17
6
31
N
IF(ERROR) ,GOTO A
A::
,R JOBCOS
18:00
,DELETE PVT3:DIS32,DAT
,DISMOUNT 16:
,DISMOUNT 17:
```

Figure 8. Sample Batch Control Files For OAGPRF



```
.MOUNT MTA1:16/RE:4342/VI:'9TRK-LAVIANA-4342'/WL
IF(ERROR) .REQUEUE
.MOUNT MTA1:17/RE:4344/VI:'9TRK-LAVIANA-4344'/WE
IF(ERROR) .REQUEUE
.REWIND 16:
.REWIND 17:
.SET BLOCKSIZE 16:2200
.SET BLOCKSIZE 17:2200
.SKIP 16:14 FILES
.SKIP 17:14 FILES
.R SETSRC
T
C PVT3:
T
.R JOBCOS
18:00
.RUN OAGPRF
73
8,1
8,31
16
17
7
17
YES
14
N
IF(ERROR) .GOTO A
A::
.R JOBCOS
18:00
.DELETE PVT3:DIS33.DAT
.DISMOUNT 16:
.DISMOUNT 17:
```

Figure 9. Sample Batch Control Files For OAGPRF

a subset of the input files can be merged providing their tags are consecutive. If more than eight months are being used it it is necessary to change the device numbers being used. Currently, devices 16-23 are used for input files, and 24 is the output device. If ITAG in OAGPRF was changed, it will be necessary to change it here as well. For each airport, PRFMRG produces a file named tlcPM.DAT. These files will be read by CLUSTR.

Next, run CLUSTR. Figure 10 provides a sample batch control file. This program also uses the list of airports, which the user may be changing. CLUSTR writes files which will be read by CLSFRQ and APMFG, and AVGPRF.

CLUSTR is currently set up to cluster 120 days. It requires about 70K core and 10 cpu minutes to cluster this many days. If the total number of days in the months being used is not 120, it will be necessary to redimension: in subroutine CLSSUB, arrays IDAY, NDC, DVOL, NAME, THRES; in subroutine HICLUS, arrays TREE, LINE, LAMBDA, OUTPUT; and in subroutine PRTREE, arrays MU, MARK, and LAMBDA. Exact dimensioning is preferred for these arrays.

If more than 120 days are being clustered it will also be necessary to change the dimensions of ARRAY in the main program, and PROX in subroutines EUCLID and HICLUS. Let NDAYS be the number of days to be clustered. Then ARRAY is dimensioned NDAYS*96 and PROX is dimensioned (NDAYS*(NDAYS-1))/2.

```
.R SETSRC
T
C PVT3:
T
.R JOBCUS
14:00
.RUN CLUSTR
NO
3
YES
2,4
YES
.R JOBCUS
14:00
```

Figure 10. Batch Control File For CLUSTR

Currently, ARRAY is dimensioned 12000 and PROX is dimensioned 8000.

After clustering is complete, AVGPRF, using disk files written by CLUSTR and tapes written by OAGPRF, begins the averaging process for profiles and related quantities. AVGPRF writes interim disk files, one per month for each airport, containing summed quantities for the parts of each cluster in the month. As AVGPRF is run for each month, specify which month is being done. Simply number the months 1, 2, 3,...up to the number of months being done. Put them in chronological order.

Array MONTHS stores the number of days in each month being used, in chronological order. Dimension MONTHS as the number of months, and NDC and PFS as the total number of days. Subroutine WRT also uses some of these arrays. If more than eight months are being used, it will be necessary to expand array NNM, which contains the tags for the disk files AVGPRF produces. See Table 7 for a list of these tags. If another number of clusters than three is being used, it will be necessary to change the dimension of everything that has a "3" in it.

Once AVGPRF has been run for each month, there will be one disk file per month for each airport. AVGMRG merges these files; any changes made in previous programs should be extended to this one. NF is the number of input files, equal to the number of months; NDAYS is the number of days. There are two subroutines, READD, which reads the files written by AVGPRF, and WRT, which writes an averaged output file. The output file is named tlcCP.DAT. Device 20 is output file is named tlcCP.DAT. Device 20 is used for output, and device 21 is used for input files. The files tlcCP.DAT will be used by APMFG, described in Section 7.5.

7.3 Weather

Weather data processing is independent of all other processing. There are three stages involved: data extraction, data compaction, and determination of VFR and IFR days.

Program EXTWEA extracts ceiling and visibility data from a TDF-14 tape. Records are read in I format and any records containing non-numeric data are reread in (20A1) format and written to the terminal. They are also written to the output tape.

To extract weather data for some other year or years, only two statements need be changed. Rewrite the statement IF (IA(2).EQ.72.OR. ...) GO TO 100 to specify the year or years desired. Similarly, rewrite the statement IF (AR(7).EQ.'2'.OR....) GO TO 500. Of course, the statement 10 FORMAT(I5,4I2,1X,2I3) could be replaced with the statement 10 FORMAT(A5,4A2,1X,2A3), and Hollerith fields put in the statement IF (...) GO TO 100. While this would have the advantage of making the extraction faster, it has the disadvantage of concealing any non-numeric data on the input tape, and some caution may be advisable in taking this step.

Once the data has been extracted for the years desired, run WEACOM. This program reads eight records at a time from the input tape; eight being the number of daily observations made in '72 and '73. If 24 daily observations are available, change LIM from 8 to 24, and dimension arrays ICLG (ceiling) and IVIS (visibility) at 24. If there is a read error, the ceiling and visibility are set to zero for that observation.

Observations for each day are written to the output tape as one logical record. An end-of-file is written to the output tape at the end of the data. A list of station numbers is written to the terminal.

There are eleven statements in the program which are special pleading for two missing records for Seattle in December 1972 December 1972. These are indicated by comments in the program list. An assumption was made that the missing observations were the last two for the last day of the month, and these were assumed equal to the last two observations for the preceding days. If desired, these statements can be removed. Note that if the the TDF-14 tape has either missing or extra records, the program will have to be modified to deal with that fact. See Section 4.0 for a fuller discussion of this topic.

The compacted weather tape is read by program VERIFR, which produces a list, for each station, of VFR and IFR days. Arrays ICL and IVS should be dimensioned at the number of observations per day. LIST should be dimensioned at the total number of days in the year or years which are being used. This is also the value assigned NDAYS, now 731. IAIR is dimensioned at the value given NA, the number of stations, currently 30. DATA IAIR/.../ should contain a list of the three-letter codes for the airports being used in the order that the corresponding stations appear on the tape. (It will be remembered that WEACOM gives a list of the station numbers.)

Currently, the IFR/VFR decision is made as follows:

There are eight observations for each day. Consider observations 3 through 8. If ceiling ≥1500 feet for 5 or more of these six observations, and visibility ≥3 miles for 5 or more of these observations, then the day is VFR. If not, it is IFR. The list of VFR/IFR days is written to disk file tlcWE.DAT as an array of 0's and 1's, 0 representing IFR weather and 1 representing VFR weather.

7.4 Tower

The first program run for tower data is TOWRD. One run of this program is needed for each 7230.1 tape. If there are tapes for more than one year, array NPT may need to be changed between years. Section 5.0 discusses the reasons for this.

In TOWRD, dimension arrays NPT and AN to the number of airports being used. DATA AN, NPT/.../ lists the three-letter codes for the airports and the corresponding positions on the 7230.1 tapes that these airports occupy. For example, BOS is the seventh airport on the 7230.1 directory for 1972, CLT is the 110th, and so forth. TOWRD3 is the same program with NPT rewritten for 1973; BOS is still 7th CLT is 113th, etc. The list needed by TOWRD can be made by simply going through the 7230.1 directory by hand, or a card deck version of the directory which is also provided with the tapes can be used.

When running these programs, in addition to entering the logical device number for the input tape, a question will be asked regarding the range of airports on the tape. Since data for 100 airports is on each tape except the last (which has as many as are left over) the answer will be '1,100', '101,200', etc., depending on which tape is mounted. It will also be necessary to specify a anme for the output file, as these programs write data for the airports extracted from a particular tape to a single output file.

Now run TOWMRG. This program should have as many input files as there are years of data. Note that TOWMRG assumes that data for a particular airport is also in the same one of the 7230.1 tapes (2 of 4, for instance). Also note that the arrays read in by this program include nine locations (the last nine) which will be empty if the year was not a leap year. In the case of 72-73, this was not a problem as the only empty locations were at the end of the '73 array.

However, if a year that is not the last one being used is not a leap year (for instance, 74-75), write out only the part of the array for that year that actually contains data, so that data for 1/1 of the second year and 12/31 of the first year will be contiguous. This means, for instance, that you would change IARRAY in the statement WRITE(22)... to (IARRAY (I), I=1,3285). Note also that TOWMRG must be run even if there is only one year of data, as it produces separate files for each airport named tlcTW.DAT. Once these files and weather files (See Section 7.3) are written, TOWER can be run.

Set NDAYS to the total number of days for tower and weather data. Dimension LIST, VS, VN to the value of NDAYS, and MIX to (8,NDAYS). Dimension NARRAY to 8*NDAYS. Dimension AN to the number of airports being used, and enter the airport list in DATA AN/.../. Set NAPT to the number of airports. In subroutine XFRDIS, dimension IORD1 and IORD2 to the number of days. Current dimensioning allows up to five distribution levels for scheduled and non-scheduled volumes, although presently only three are used. These levels are in data statements in XFRDIS, and are currently set to .5, .85, 1.0 for both scheduled and non-scheduled, VFR and IFR. <u>TOWER</u> produces two output files for each airport: tlcTO.DAT and tlcIV.DAT. If TOWER is to be run for a subset of the list of airports, it will be necessary to type in the three-letter codes.

7.5 Annualization

The annualization happens in two stages: calculation of cluster frequencies, and generation of APM files. First run CLSFRG. Here, NDAYS is the number of OAG days. Dimension NDC, VOL and SVOL at this number. Dimension AP at the number of airports and insert the list of airports. CLSFRQ uses input files tlcCL.DAT, tlcIV.DAT, tlcTO.DAT and writes file tlcCF.DAT.

Next, run APMFG. Here also, NDAYS is the number of OAG days. Dimension TOT and PFS to this number. IAN contains the airport list this time. Input files are tlcCP.DAT, tlcTO.DAT, and tlcCF.DAT. Output files are tlcPF.DAT (profile file), tlcDF.DAT (daily file), tlcAF.DAT (annualization file), and tlcHF.DAT (header file). These are the final product. Enjoy them.

8. INCIDENTAL PROGRAMS AND DIVERSIONS

To provide data for APM delay validation, pseudo-daily files were required for May 1972 for Newark, LaGuardia, and JFK (EWR, LGA, JFK). TOWMAY provides tower data required on files tlcDl.DAT. Program DAILY reads these files. It also reads an OAGPRF output tape, on which the three airports are contiguous, and an hourly weather file for May 1972, tlcWP.DAT. It then writes to tape a "daily" file for each day of May 1972 for each airport. These "daily" files have the same form and content as the daily files described in Table 8. The weather files were produced by programs WEATHR and WEATH2, by a process of redefining the eight observations available to 24 "observations". Program JACK provides a printout of as many days as are desired from a file on the "daily" tape.

In additon, program WEATH4 prints out ceiling and visibility for the 31 days of May 1972. This program reads from the tape written by WEACOM, regarding which, see Sections 4.0 and 7.3.

9. PROGRAM DESCRIPTIONS

This section provides descriptions of programs discussed in this document.

9.1 Program OAGRD

This is a FORTRAN program which reads files from an OAG tape, extracts arrival and departure records for a specified list of airports, and writes these records to tape.

Input parameters are:

NTAPE - number of files on input tape

NA - number of airports

AN - list of three-letter codes; read in only if NA is not equal to the number of airports in the list contained in the program

IDEV - input device number

IOUT - output device number

A record count is written to device five every 5000 records read; the cumulative number of records extracted is written out at the end of each input file, and the total number of records read and records extracted is also written out.

9.2 Program OAGRD2

This is a FORTRAN program which reads a tape written by OAGRD and orders the records by airport.

Input parameters are:

NA - same as in OAGRD

AN - same as in OAGRD

IDEV - same as in OAGRD

IOUT - same as in OAGRD

A record count is written to file FOR21.DAT, and to device five.

9.3 Program OAGPRF

This is a FORTRAN program which reads a tape written by OAGRD2 and generates minute-by-minute profiles of arrivals and departures. Records with FLAG or FUEL for times are ignored. Quarter-hour profiles are written to disk, unnormalized. Tape output includes airport, number of days and, for each day: day-of-year, minute-by-minute profile of arrivals and departures, hourly arrival/departure ratios, concentrations and mix, and peaking factor and total daily volume. A temporary disk;file is written containing information from the OAG records. Optional printed output is available, but desirable only for small test runs (a few days).

The program asks for the following input parameters:

YEAR - year; integer - eg., 74 or 1974 →1973

DATE - month, day at which profile generation begins; for instance, if the input tape is for April, 1975, this could be 4,7

DATE - month, day of last day, e.g. 4,12.
If the whole month's profiles are wanted, then first DATE read would be 4,1, and second DATE read would be 4,30.

NJAN1 - if YEAR is not between 1970 and 1976, NJAN1 is read in - this is the day of the week that January 1 fell on in YEAR, e.g. Wednesday.

IDEV - input device number

ITAPE - output device number

NTAG - number of tag for disk file; NTAG is between 1 and 8; consecutive tags should be used, and tags should be assigned uniquely to a month. A list of the tags associated with each value of NTAG is written to device five before NTAG is read.

NA - same as in OAGRD

REP - if NA is not equal to the number of airports in the list, the program asks: "Are you just skipping some files?" Answer YES or NO.

ISKIP - number of airports being skipped; needed if NA was not maximum value and REP was YES.

AN - as in OAGRD; this list is read if NA was not maximum and REP was NO.

REPLY - printing option; = ALL, SOME or NONE; except for test runs of a few days, NONE is the best bet.

The day-of-week of the first day wanted is written to device five, as is the name of each airport as profile generation for it is begun. Records which are ignored are also written to device five. If the input tape has records which are not for the expected airport, the message 'you are somewhat mistaken as to the information on this tape' is written to device five.

9.4 Program PRFMRG

This is a FORTRAN program which merges, for all or some of a list of airports, up to eight disk files containing quarter-hour profiles. These disk files are produced by OAGPRF (see description above).

Input parameters are:

NA1,NA2 - locations in list of first and last airports whose disk files are being merged; e.g., 1,1 merges files for just the first airport on the list; 2,5 merges files for the next four airports; 1,31 merges files for all of a list of 31 airports.

NF1,NF2 - numbers of first and last files to be merged - these correspond to the lowest and highest values specified for NTAG in runs of OAGPRF (see above).

The range of days on each input file is written to device five.

9.5 Program CLUSTR

This is a FORTRAN program which performs hierarchical proximity clustering on quarter-hour profiles contained in files written by PRFMRG. An alternate version, PCLUST, will cluster for any input file, the file name being read in.

Input parameters for CLUSTR are:

REPLY - NO if trees are not to be printed; otherwise, anything.

NC - number of clusters wanted

REPLY - YES if trees are to be saved on disk, which is a good idea.

NA1,NA2 - same as for PRFMRG

REPLY - NO if normalized profiles are not wanted, usual answer is YES.

9.5.1 Subroutine CLSSUB

This is a FORTRAN subroutine which normalizes input data, if required, and calls cluster software.

The calling sequence is:

```
CALL CLSSUB(AIR,ARRAY,NROWS,NCOLS)
```

where the arguments of the subroutine are:

AIR - an integer variable containing a three-letter airport code

ARRAY - array of profiles, dimensioned NROWS x NCOLS

NROWS - rows in ARRAY; number of days being clustered

NCOLS - columns in ARRAY; here, 96 (quarter-hours/day)

9.5.2 Subroutines EUCLID, HICLUS, PRTREE

These FORTRAN routines are described in detail in report KHL-TSC-76-1396, dated April 20, 1976.

9.5.3 Subroutine SUSPCL

This FORTRAN subroutine suspends the cluster process at any number of clusters and produces a list of the assignments of days to clusters. With current dimensioning of INDEX, the maximum number of clusters is 10.

The calling sequence is:

```
CALL SUSPCL(NAME,THRES,NDC,NUM)
```

where the arguments of the subroutine are:

NAME,THRES - as for cluster software, op. cit.

NDC - for day I, NDC(I) is the cluster assignment.

NUM - number of days.

If the number of clusters is three, a default subroutine, YAWN, is called.

9.5.4 Subroutine YAWN

This FORTRAN routine suspends clustering at three clusters.

The calling sequence is:

```
CALL YAWN(NAME,THRES,NOC,NUM)
```

where the subroutine arguments are the same as those for SUSPCL.

9.6 Program PRFPLT

This is a FORTRAN program which produces terminal plots of quarter-hour profiles, daily or clustered.

Input parameters are:

NAME - name of input file

YEAR - year

NA - number of airports on input file

REPLY - NO if normalized data is not wanted

9.6.1 Subroutine PLTSUB

This FORTRAN routine is exactly analogous to CLSSUB above. Instead of calling cluster software, it calls XYPLOT.

Input parameters are:

THIS - YES if clusters are wanted; NO if individual days are to be plotted.

if THIS was 'YES':

NPLT - number of days to be plotted; current maximum 10.

IPLT - array; numbers of days

if THIS was 'NO':

NCLUS - number of clusters

DEF - YES if cluster default is to be used, which puts days 1-29 in cluster 1, days 30-91 in cluster 2, and days 92-241 in cluster 3. This option is probably not useful.

if DEF was 'NO', for each cluster:

NDJ - number of days in cluster - maximum is 241

NDJA - which days they are.

9.6.2 Subroutine XYPLOT

This subroutine produces the plots. The calling sequence is:

```
CALL XYPLOT(XMIN,XMAX,XDEL,Y)
```

where the subroutine arguments are:

XMIN - minimum value of X - here, 1.

XMAX - maximum value of X - here, 96.

XDEL - stepsize - here, 1.

Y - array of Y values.

Input parameters for the subroutine are:

NCH - number of characters in title

TITLE - title.

9.7 Program CORAL

This is a FORTRAN program which provides a set of correlations between clusters for February 1975 and three months of 1975, and between days of the three month period and February clusters, for five airports. The clusters are read from file CORAL.DAT.

9.7.1 Subroutine CCOEFF

This subroutine calculates a correlation coefficient, defined by

$$r = \frac{\sum_{i=1}^{n} X_i \, Y_i - \frac{\Sigma X_i \, \Sigma Y_i}{n}}{\left[\left(\Sigma X_i^2 - \frac{(\Sigma X_i)^2}{n}\right)\left(\Sigma Y_i^2 - \frac{(\Sigma Y_i)^2}{n}\right)\right]^{1/2}} .$$

The calling sequence for this subroutine is:

CALL CCOEFF(N,X,Y,R)

where the arguments are defined as

N - n in the above equantion

X - X above; a profile

Y - Y above; a profile

R - r above; the correlation coefficient.

9.7.2 Subroutine SUM

This subroutine, called from CCOEFF, sums the elements of an array. The calling sequence is:

CALL SUM(X,N,S)

where the subroutine arguments are

X - array

N - number of elements being summed.

S - sum.

9.8 Program AVGPRF

This is a FORTRAN program which performs partial averaging for profiles and ancillary quantities, read from a tape written by OAGPRF. Input parameters are:

IDEV - input device number

NA1,NA2 - same as for PRFMRG

ITAPE - an integer which specifies tags for output files. It may vary between NF1 and NF2 as defined for PRFMRG.

NT1 - The smallest value of ITAPE; may equal NF1 in PRFMRG

9.9 Program AVGMRG

This is a FORTRAN program which merges the files written by AVGPRF, completing the averaging process.

Input parameters for this program are:

NA1,NA2 - as in previous programs

NF - number of files being merged

NF1 - tag of first file; same as NT1 in AVGPRF

9.10 Programs TOWRD and TOWRD3

These are FORTRAN programs which read 7230.1 tapes and extract data for selected airports. TOWRD does this for 1972;

TOWRD3 for 1973. Output is to disk. Input parameters are:

ANAME - name for output file

ITAPE - input device number

NAT1,NAT2 - range of airports on the input tape, e.g., if this tape has the second hundred airports, as listed in the 7230.1 directory, the NAT1,NAT2 = 101,200.

9.11 Program TOWMRG

This is a FORTRAN program which merges data for two files, one written by TOWRD and one written by TOWRD3, and writes a separate disk file for each airport for which there is data on the two input files.

Input parameters are:

NAME1,NAME2 - names of two input files

NA - number of airports on the input files.

9.12 Program TOWER

This is a FORTRAN program which separates tower traffic by weather; calculates the non-scheduled mix for each day; computes scheduled vs. non-scheduled distributions for VFR and IFR days, associates mean volumes and mean mixes with these distributions; and calculates over-all means for air carrier, air taxi, general aviation and military traffic.

Input parameters are:

NA - number of airports

IAIR - name of an airport; the program prompts for this if NA is smaller than the number of airports listed.

Major variables are:

VS - array of daily scheduled volumes

VN - array of daily non-scheduled volumes

MIX - array of daily mixes

9.12.1 Subroutine XFRDIS

This is a FORTRAN subroutine which, for two subsets of days (VFR or IFR), calculates scheduled and non-scheduled distribution levels, and defines a set of "boxes" limited by these levels. It also averages scheduled and non-scheduled volumes and mixes for each of the boxes, calculates intervals I and J such that I(i) is the largest scheduled volume for the i^{th} set of boxes, and J(j) is the largest non-scheduled volume for the j^{th} set of boxes.

The calling sequence for this subroutine is:

```
CALL XFRDIS(NDAYS,VS,VN,MIX,LIST,MNUM)
```

where the arguments are defined as:

NDAYS - total number of days

VS - array of scheduled volumes, dimensioned NDAYS

VN - array of non-scheduled volumes, dimensioned NDAYS

MIX - array of mixes, dimensioned 8 x NDAYS.

LIST - list of VFR/IFR days, dimensioned NDAYS

MNUM - array of two elements containing number of IFR days and number of VFR days

Other major variables are:

NPI - number of scheduled volume distribution levels; two elements (IFR/VFR)

NSIG - number of non-scheduled volume distribution levels; two elements (IFR/VFR)

PI - array of scheduled distribution levels (fractions)

SIG - array of non-scheduled distribution levels

ILIST - integer keying IFR/VFR calculations, if 0, IFR; if 1, VFR

9.12.2 Subroutine DISLEV

This is a FORTRAN subroutine which calculates the number of days below each distribution level. The calling sequence for this subroutine is:

```
CALL DISLEV(N,AR,X,NAR)
```

where the arguments are defined as:

N - number of distribution levels; NPI or NSIG

AR - PI or SIG in XFRDIS; dimensioned 0/1 x N

X - number of days; floating point

NAR - returns the number of days below each level; dimension N.

9.12.3 Subroutine ORDER

This is a FORTRAN subroutine which produces a list of array subscripts ordered by increasing value of array elements for a subset of an array. The calling sequence for this routine is:

```
CALL ORDER(ND,V,LIST,IC,IORD)
```

where the arguments are defined as:

ND - number of days (total)

V - array of volumes

LIST - list of VFR/IFR days

IC - number of days in the subset (either VFR or IFR)

IORD - list of subscripts of subset of volumes, ordered by increasing volume.

9.13 Program CLSFRQ

This is a FORTRAN program which calculates the frequency with which each cluster occurs for each scheduled distribution level. Input parameters are:

NA1,NA2 - same as in PRFMRG

Subroutines called by CLSFRQ are DISLEV and ORDER, which differ from their versions in TOWER only in that keying for IFR/VFR days has been removed.

9.14 Program APMFG

This is a FORTRAN program which reads files of tower data, clustered profiles and cluster frequencies; and produces, for use by the Airport Performance Model, header, annualization, profile, and daily files. Input parameters are:

NA1,NA2 - same as in PRFMRG

9.15 Program FPRINT

This is a FORTRAN program which provides printing of any of the files written by APMFG. Input parameters are:

IAIR - Three-letter code for airport

A - D - for daily file

H - for header file

A - for annualization file

P - for profile file

N - or any other letter, end run.

APPENDIXES TO PART I

APPENDIX A

```
***********************************************************************
*                    OAG PROFILE PRODUCTION PROGRAMS
***********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:OAGRD.F4  [4255,516] 18-May-76
***********************************************************************.
C                 OAGRD.F4
C
C        THIS PROGRAM READS ONE OR MORE FILES ON AN OAG TAPE AND EXTRACT.
C        ARRIVAL AND DEPARTURE RECORDS FOR A SPECIFIED LIST OF
C        AIRPORTS.   THESE RECORDS ARE WRITTEN TO TAPE IN THE SAME
C        ORDER AS THEY ARE READ.
C
      DIMENSION ALIST(22),AN(31),ANS(31)
      DATA ANS/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
      WRITE(5,1100)
 1100 FORMAT('  HOW MANY FILES ON INPUT TAPE?'/)
      READ(5,200) NTAPE
      IREAD=0
      IWRITE=0
      WRITE(5,100)
  100 FORMAT('  HOW MANY AIRPORTS?'/)
      READ(5,200) NA
  200 FORMAT(I)
C
C           IF NOT ALL THIRTY-ONE AIRPORTS ARE BEING EXTRACTED,
C           CODES FOR THOSE WANTED MUST BE TYPED IN
C
      IF(NA.NE.31) GO TO 110
      DO 111 I=1,NA
  111 AN(I)=ANS(I)
      GO TO 112
  110 WRITE(5,300)
  300 FORMAT(' NAME AIRPORTS - 3 LETTER CODES - N(A3,1X) '/)
      READ(5,400) (AN(J),J=1,NA)
  400 FORMAT(30(A3,1X))
  112 CONTINUE
      WRITE(5,500)
  500 FORMAT(' INPUT TAPE DEVICE NUMBER? '/)
      READ(5,200) IDEV
      WRITE(5,700)
  700 FORMAT('  OUTPUT TAPE DEVICE NUMBER? '/)
      READ(5,200) IOUT
  800 FORMAT(I2,1X,A5)
      DO 6000 ITAPE=1,NTAPE
      WRITE(5,1300) ITAPE
 1300 FORMAT(///'  ITAPE = ',I2)
  210 READ(IDEV,600,END=1000,ERR=2000) ALIST
  600 FORMAT(A4,A3,2A5,A4,A3,15A5,A3)
      IREAD=IREAD+1
      IF(IREAD/5000*5000.EQ.IREAD) WRITE(5,770) IREAD
  770 FORMAT('  RECORDS = 'I8)
      JA=0
  310 JA=JA+1
      IF(JA.GT.NA) GO TO 210
      IF(ALIST(2).NE.AN(JA).AND.ALIST(6).NE.AN(JA)) GO TO 310
      WRITE(IOUT,600) ALIST
      IWRITE=IWRITE+1
      GO TO 210
 2000 IREAD=IREAD+1
      WRITE(5,2001) IREAD
 2001 FORMAT('  READ ERROR AT RECORD 'I8/)
      GO TO 210
 1000 CONTINUE
      WRITE(5,1200) ITAPE,IWRITE
 1200 FORMAT(/'  FOR FILE:',I2,', RECORDS EXTRACTED = 'I8)
      WRITE(5,900) IREAD,IWRITE
  900 FORMAT(//I10,' RECORDS READ FROM TAPE'/' WRITTEN TO TAPE:',I10,
     1' RECORDS'/)
 6000 CONTINUE
      END FILE IOUT
      END
```

```
********************************************************************************
*                        OAG PROFILE PRODUCTION PROGRAMS
********************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:OAGRD2.F4 [4255,516] 18-May-76
********************************************************************************
C                   OAGRD2.F4
C
C               THIS PROGRAM READS TAPE(S) WRITTEN BY OAGRD.F4 AND
C       ORDERS THE RECORDS BY AIRPORT FOR THE AIRPORTS SPECIFIED.
C       NOT ALL THE AIRPORTS EXTRACTED BY OAGRD.F4 NEED BE SPECIFIED
C       HERE.  THE CONVERSE, HOWEVER, IS NOT TRUE.  A RECORD COUNT IS
C       WRITTEN ON FOR21.DAT.
C
        DIMENSION ALIST(22),AN(31),ANS(31)
        DATA ANS/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
       1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
       2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
       3 'LAX','SFO','LAS','SEA','HNL'/
        ITAPE=1
        IREAD=0
        IWRITE=0
        WRITE(5,100)
    100 FORMAT('  HOW MANY AIRPORTS?'/)
        READ(5,200) NA
    200 FORMAT(I)
C
C             IF NOT ALL THIRTY-ONE AIRPORTS ARE BEING EXTRACTED,
C             CODES FOR THOSE WANTED MUST BE TYPED IN
C
        IF(NA.NE.31) GO TO 110
        DO 111 I=1,NA
    111 AN(I)=ANS(I)
        GO TO 112
    110 WRITE(5,300)
    300 FORMAT(' NAME AIRPORTS - 3 LETTER CODES - N(A3,1X) '/)
        READ(5,400) (AN(J),J=1,NA)
    400 FORMAT(30(A3,1X))
    112 CONTINUE
        WRITE(5,500)
    500 FORMAT(' INPUT TAPE DEVICE NUMBER? '/)
        READ(5,200) IDEV
        WRITE(5,700)
    700 FORMAT('  OUTPUT TAPE DEVICE NUMBER? '/)
        READ(5,200) IOUT
    800 FORMAT(I2,1X,A5)
        DO 5000 JA=1,NA
        IB=0
        IA=0
        REWIND IDEV
    210 READ(IDEV,610,END=1000,ERR=2000) ALIST(2),ALIST(6)
```

```
  610 FORMAT(4X,A3,14X,A3)
  600 FORMAT(A4,A3,2A5,A4,A3,15A5,A3)
      IB=IB+1
      IREAD=IREAD+1
      IF(IREAD/10000*10000.EQ.IREAD) WRITE(5,770) IREAD
  770 FORMAT('  RECORDS = 'I8)
      IF(ALIST(2).NE.AN(JA).AND.ALIST(6).NE.AN(JA)) GO TO 210
      REREAD 600,ALIST
      WRITE(IOUT,600) ALIST
      IA=IA+1
      IWRITE=IWRITE+1
      GO TO 210
 2000 IREAD=IREAD+1
      IB=IB+1
      WRITE(5,2001) IREAD
 2001 FORMAT('  READ ERROR AT RECORD 'I8/)
      GO TO 210
 1000 CONTINUE
      END FILE IOUT
      WRITE(21,880) AN(JA),IA,IB
      WRITE(5,880) AN(JA),IA,IB
  880 FORMAT(6X,'  FOR ',A3,', RECORDS EXTRACTED ='I10/10X,
     1 'RECORDS READ ='I10)
 5000 CONTINUE
      WRITE(5,900) IREAD,IWRITE
  900 FORMAT(//I10,' RECORDS READ FROM TAPE'/' WRITTEN TO TAPE:',I10,
     1' RECORDS'/)
 6000 CONTINUE
      END FILE IOUT
      END
```

```
**********************************************************************
*                      OAG PROFILE PRODUCTION PROGRAMS
**********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:OAGPRF.F4 [4255,516] 18-May-76
**********************************************************************
C                      OAGPRF.F4
C
C              THIS PROGRAM READS A TAPE WRITTEN BY OAGRD2.F4 AND
C        GENERATES MINUTE-BY-MINUTE PROFILES OF OPERATIONS.
C           AIRPORTS ARE CONSIDERED IN THE ORDER IN WHICH THEY
C        OCCUR ON THE INPUT TAPE.    THE PROGRAM ALWAYS PRODUCES
C       TAPE(MINUTE-BY-MINUTE) AND DISK(QUARTER-HOURLY) OUTPUT AND
C       HAS OPTIONAL PRINTED OUTPUT.    BOTH TAPE AND DISK OUTPUT FILES
C       INCLUDE AIRPORT, NUMBER OF DAYS, AND, FOR EACH DAY, DAY-OF-YEAR,
C       ARRIVAL/DEPARTURE RATIOS, CONCENTRATIONS MIX PERCENTAGES,
C          AND PEAKING FACTORS.
C
        DIMENSION EQCD(172),MNTH(12),AN(31),ANS(31),NUMA(31),
       1 VOL(1440,2),DATE(2),A(2),TIME(2,2),DAYS(7),DATES(4),DAY(2),
       2 DY(2),MINAD(2),HOUR(2),ADR(24),VOLQ(96,2),VOLH(24,2),
       3 AMX(4),CONCEN(24),IMS(4),WEEK(7),JAN(7),
       4 WQ(96),WM(1440),WH(24),PMIX(24,8),PMIXP(24,8)
        LOGICAL IPRINT,IPRNT
        INTEGER PCL(172)
        INTEGER TIME,DAYS,DATES,YEAR,DATE,DAY,DY,DOFW,EQCL,
       1HOUR,DOFW1,EQC,AN,A,ANS
        DIMENSION ITAG(8)
        DATA ITAG,IAND/"31142,"31144,"31146,"31150,
       1 "31542,"31544,"31546,"31550,"777777700000/
        DATA WEEK/'SUN','MON','TUE','WED','THU','FRI','SAT'/
        DATA JAN/5,6,7,2,3,4,5/
        DATA IPRINT,IPRNT/2*.FALSE./
        DATA MNTH/31,28,31,30,31,30,31,31,30,31,30,31/
        DATA NOTHP,NCLP/7,8/
        DATA PCL/1,2*3,6*2,3,1,7*2,3*4,5,4,2,5,5,7*4,
       1 2*2,2*4,5,5*4,5*5,7*4,5,4,4,2,2,4,5,65*6,2*6,
       2 25*6,18*8/
        DATA EQCD/'747','D10','L10','D8S',
       1 'D8F','B3F','Y62','DC8','707','A3B','74F','SSC','Y76','GAL',
       2 'T44','STV','V10','SUV',
       3 '111','BAC','S11','72S','737','720','727','72F','73S',
       4 'VL3','VL6','CVS','CVF','CVL','J60','880','990','FAL',
       5 'DME','CM4','DC9','D9S','D9F','FFJ','F28','TRD','T1C',
       6 'T1E','T2E','T3B','HAN','HSJ','LJT','T24','T04','T34','VF6',
       7 'Y40','JET','D95','B2F','B7F','J6 ','T54',
       9 'AVS','AVR','AV ','MER','ME ','VAN','BRF','C44','CL ','CV2',
       A 'C2 ','CV3','C3 ','CV4','C4 ','CV5','C5 ','CV6','C6 ','CVR',
       8 'A12','A1 ','A2 ','A24','A02','AN ','A10','A22','A26','ARE',
       1 'CV ','PCU','CU ','D6V','B2 ','PDS','DS ','DHC','DH7',
       2 'DC3','D3 ','DC4','D4 ','DC6','D6 ','D6A','6A ','D6F','D6B',
     3 '6B ','D6C','DC7','FPR','F27','F7 ','FH7','FH ','PGU','GU ',
     4 '748','A0 ','Y14','Y4 ','Y18','Y8 ','LEC','LE ','LHE','LXA',
     5 'PHR','HR ','MR4','M4 ','Y51','YS ','N26','N2 ','NAT','T14',
     6 'T4 ','VIS','VV ','VE ','V70','V80','V8 ','YK2','Y11',
     B 'SU ','A5 ','CO ','N2 ',
     7 'ALO','AL ','AB4','AB2','205','SA3','VT7','HOV','KH4',
     8 'S34','S61','S1 ','S55','S58','PS4','S4 ','47J','JR '/
      DATA ANS/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
C
C          GET INITIAL INFORMATION
C
      WRITE(5,100)
  100 FORMAT(' WHAT YEAR ARE WE IN?'/)
      READ(5,101) YEAR
  101 FORMAT(2I)
      IF(YEAR.LT.1900) YEAR=YEAR+1900
      IF(YEAR/4*4.EQ.YEAR) MNTH(2)=29
      DO 80 LD=1,2
      IF(LD.EQ.1) WRITE(5,102)
      IF(LD.EQ.2) WRITE(5,105)
  102 FORMAT('  WHAT IS THE FIRST DAY WE WANT - MONTH,DAY?'/)
      READ(5,101) DATE
      IF(LD.EQ.1) IDS1=DATE(2)-1
      DAY(LD)=0
      IF(DATE(1).EQ.1) GO TO 200
      N=DATE(1)-1
      DO 201 JD=1,N
  201 DAY(LD)=DAY(LD)+MNTH(JD)
  200 DAY(LD)=DAY(LD)+DATE(2)
  105 FORMAT(' WHAT IS THE LAST DAY WE WANT? - MONTH,DAY'/)
   80 CONTINUE
      NDAYS=DAY(2)-DAY(1)+1
C
C    FIND DAY-OF-WEEK FOR FIRST DAY WANTED
C
      IF(YEAR.LT.1970.OR.YEAR.GT.1976) GO TO 1515
      IDW=YEAR-1969
      JAN1=JAN(IDW)
      GO TO 1616
 1515 WRITE(5,1717) YEAR
 1717 FORMAT('  SORRY - I DON',1H','T KNOW WHAT DAY OF THE WEEK'/
     1 ' JANUARY FIRST OF 'I4,' WAS -'/
     2 ' YOU WILL HAVE TO TELL ME'/)
      READ(5,1818) NJAN1
 1818 FORMAT(A3)
      DO 1919 IJ=1,7
 1919 IF(WEEK(IJ).EQ.NJAN1) JAN1=IJ
 1616 CONTINUE
      I=DAY(1)-1
      IADD=MOD(I,7)
      JAN1=JAN1+IADD
      DOFW=MOD(JAN1,7)
      IF(DOFW.EQ.0) DOFW=7
      WRITE(5,106) WEEK(DOFW)
  106 FORMAT('  THE FIRST DAY YOU WANT FALLS ON 'A3)
      DOFW=DOFW-1
C
C  GET INPUT AND OUTPUT TAPE DEVICE NUMBERS AND NAME OF DISK OUTPUT FIL
C
      WRITE(5,301)
  301 FORMAT(' INPUT TAPE DEVICE NUMBER?'/)
      READ(5,101) IDEV
      WRITE(5,302)
  302 FORMAT('  OUTPUT TAPE DEVICE NUMBER?'/)
      READ(5,101) ITAPE
      IOUT=21
      WRITE(5,3301)
 3301 FORMAT('   SELECT TAG FOR OUTPUT DISK FILE NAMES:'/
    1 '     ENTER AN INTEGER BETWEEN 1 AND 8'/
    2 '       1: 21 (FEB 72)'/
    3 '       2: 22 (MAY 72)'/
    4 '       3: 23 (AUG 72)'/
    5 '       4: 24 (NOV 72)'/
    6 '       5: 31 (FEB 73)'/
    7 '       6: 32 (MAY 73)'/
    8 '       7: 33 (AUG 73)'/
    9 '       8: 34 (NOV 73)'/)
      READ (5,101) NTAG
      JTAG=ITAG(NTAG)
      IDSK=20
      WRITE(5,70)
   70 FORMAT(' HOW MANY AIRPORTS?'/)
      PEAD(5,101) NA
C
C     IF NOT ALL THIRTY-ONE AIRPORTS ARE BEING USED, CODES FOR THOSE
C         WANTED ARE TYPED IN
C
      IF(NA.NE.31) GO TO 71
      DO 72 I=1,NA
   72 AN(I)=ANS(I)
      GO TO 73
  71 CONTINUE
     WRITE(5,174)
 174 FORMAT('  ARE YOU JUST SKIPPING SOME AIRPORTS?'/)
     READ(5,81) REP
     IF(REP.EQ.'Y') GO TO 176
     WRITE(5,74)
  74 FORMAT(' NAME AIRPORTS - 3 LETTER CODES - N(A3,1X)'/)
     READ(5,75) (AN(J),J=1,NA)
  75 FORMAT(15(A3,1X))
     GO TO 73
 176 WRITE(5,177)
 177 FORMAT('    HOW MANY FILES ARE YOU SKIPPING?'/)
     READ(5,101) ISKIP
     DO 172 I=1,NA
 172 AN(I)=ANS(I+ISKIP)
  73 CONTINUE
  77 FORMAT(30I)
     WRITE(5,84)
  84 FORMAT('  FOR PRINTED OUTPUT -'/
    1 '    DO YOU WANT ALL, SOME OR NONE?'/)
     READ(5,81) REPLY
  81 FORMAT(A1)
     IF(REPLY.EQ.'A') IPRINT=.TRUE.
     IF(REPLY.EQ.'S') IPRNT=.TRUE.
     IA=0
 500 LIMIT=0
 506 IA=IA+1
     IF(IA.GT.NA) GO TO 505
     IF(IA.LE.1) GO TO 653
     END FILE ITAPE
     CALL RELEAS(IOUT)
 653 NAME=(AN(IA).AND.IAND).OR.JTAG
     CALL OFILE(IOUT,NAME)
     WRITE(ITAPE) AN(IA),NDAYS
     WRITE(IOUT) AN(IA),NDAYS
 701 FORMAT(' AIRPORT : ',A3)
     REWIND IDSK
C
C    OPEN TEMPORARY DISK FILE FOR WRITING
C
     N73='DIS'
     NDSK=(N73.AND.IAND).OR.JTAG
     CALL OFILE(IDSK,NDSK)
     WRITE(5,701) AN(IA)
C
C    READ A RECORD FROM INPUT TAPE
C
```

```
  502 READ(IDEV,501,ERR=4711,END=4502) A(1),TIME(1,1),TIME(1,2),
     1 A(2),TIME(2,1),
     1 TIME(2,2),EQCODE,DAYS,ISUPP,DATES
  501 FORMAT(4X,A3,2X,2I2,8X,A3,2X,2I2,6X,A3,13X,7I1,I1,5X,4I2)
      IF(ISUPP.EQ.1) GO TO 502
      IF(A(1).NE.AN(IA).AND.A(2).NE.AN(IA)) GO TO 507
C
C         GET POLLUTION CLASS FROM EQUIPMENT CODE
C
      IEQ=0
  601 IEQ=IEQ+1
      IF(IEQ.GT.172) GO TO 604
      IF(EQCODE.EQ.EQCD(IEQ)) GO TO 602
      GO TO 601
  602       IPC=PCL(IEQ)
      GO TO 605
  604       IPC=NOTHP
  605 CONTINUE
C
C       CONVERT EFFECTIVE AND DISCONTINUE DATES TO DAY-OF-YEAR
C
      DO 69 LD=1,2
      DY(LD)=0
      ID2=2*LD
      ID=ID2-1
      IF(DATES(ID).EQ.0) GO TO 61
      IF(DATES(ID).EQ.1) GO TO 65
      N=DATES(ID)-1
      DO 66 JD=1,N
   66 DY(LD)=DY(LD)+MNTH(JD)
   65 DY(LD)=DY(LD)+DATES(ID2)
      GO TO 69
   61 DY(LD)=DAY(LD)
   69 CONTINUE
C
C     CONVERT ARRIVAL AND DEPARTURE TIMES TO MINUTES OF DAY
C
      DO 85 I=1,2
      HOUR(I)=TIME(I,1)
   85 MINAD(I)=TIME(I,1)*60+TIME(I,2)
C
C    WRITE A RECORD TO TEMPORARY DISK FILE
C
C
C            LIMIT IS A RECORD COUNTER
C
      LIMIT=LIMIT+1
      WRITE(IDSK) A,HOUR,MINAD,IPC,DAYS,DY
 5502 GO TO 502
 4502 CONTINUE
      IDS=IDS1
      DOFW1=DOFW
      ICNT=0
      IDAY=DAY(1)-1
C
C         HAVE WE FINISHED GETTING INPUT FOR THE PROFILES
C              FOR THE CURRENT AIRPORT?
C
  550 ICNT=ICNT+1
      IF(ICNT.GT.NDAYS) GO TO 500
C
C     REWIND TEMP DISK FILE AND OPEN FOR READING
C
  811       REWIND IDSK
      CALL IFILE(IDSK,NDSK)
      DOFW1=DOFW1+1
      IF(DOFW1.EQ.8) DOFW1=1
      IDS=IDS+1
      IDAY=IDAY+1
C
C     INITIALIZE PROFILE ARRAYS - MINUTE, QUARTER-HOUR, AND HOUR
C
      DO 130 IN=1,2
      DO 131 I1=1,1440
  131 VOL(I1,IN)=0.
      DO 132 I1=1,96
  132 VOLQ(I1,IN)=0.
      DO 133 I1=1,24
  133 VOLH(I1,IN)=0.
  130 CONTINUE
      DO 134 I1=1,24
      ADR(I1)=0.
      CONCEN(I1)=0.
      DO 134 I2=1,8
      PMIX(I1,I2)=0.
      PMIXP(I1,I2)=0.
  134 CONTINUE
      DO 551 IREC=1,LIMIT
C
C     READ A RECORD FROM DISK
C
      READ(IDSK) A,HOUR,MINAD,IPC,DAYS,DY
C
C      IS THIS RECORD AN ARRIVAL OR A DEPARTURE??
```

```
C
      IAD=1
      IF(A(1).NE.AN(IA)) IAD=2
C
C     DID IT  PERATE TODAY???
C
      IF(DY( ).GT.IDAY) GO TO 551
      IF(DY(2).LT.IDAY) GO TO 551
C
C     ADD OPERATION TO APPROPRIATE MINUTE
C
      MINUTE=MINAD(IAD)+1
      ADD=FLOAT(DAYS(DOFW1))
      VOL(MINUTE,IAD)=VOL(MINUTE,IAD)+ADD
      IHOUR=HOUR(IAD)+1
C
C       ADD OPERATION TO APPROPRIATE POLLUTION CLASS, BY HOUR
C
      PMIX(IHOUR,IPC)=PMIX(IHOUR,IPC)+ADD
  551 CONTINUE
C
C       DO SOME ADDITIONAL CALCULATIONS
C
C
C     GET POLLUTION FRACTIONS
C
      DO 40 J1=1,24
      TOT=0.
      DO 241 J2=1,NCLP
  241 TOT=TOT+PMIX(J1,J2)
      DO 242 J2=1,NCLP
      IF(TOT.LT.1.0E-04) GO TO 242
      PMIXP(J1,J2)=PMIX(J1,J2)/TOT
  242 CONTINUE
   40 CONTINUE
C
C       GET QUARTER-HOUR PROFILES
C
      DO 45 JQ=1,96
      MIN1=(JQ-1)*15+1
      MIN2=MIN1+14
      DO 46 JM=MIN1,MIN2
      DO 46 J2=1,2
   46 VOLQ(JQ,J2)=VOLQ(JQ,J2)+VOL(JM,J2)
   45 CONTINUE
C
C     GET HOURLY PROFILES AND ARRIVAL/DEPARTURE RATIO
```

```
C
      DO 47 JH=1,24
      IQ1=(JH-1)*4+1
      IQ2=IQ1+3
      DO 48 JQ=IQ1,IQ2
      DO 48 J2=1,2
         VOLH(JH,J2)=VOLH(JH,J2)+VOLQ(JQ,J2)
   48 CONTINUE
      IF(VOLH(JH,2).LT.1.0E-04) GO TO 47
      ADR(JH)=VOLH(JH,1)/VOLH(JH,2)
   47 CONTINUE
C
C      SUM ARRIVALS AND DEPARTURES FOR MINUTES, QUARTER-HOURS,
C            AND HOURS
C
      DO 20 I1=1,1440
   20 WM(I1)=VOL(I1,1)+VOL(I1,2)
      DO 21 I1=1,96
   21 WQ(I1)=VOLQ(I1,1)+VOLQ(I1,2)
      DO 22 I1=1,24
   22 WH(I1)=VOLH(I1,1)+VOLH(I1,2)
C
C        GET HOURLY CONCENTRATIONS
C
      DO 30 IH=1,24
      MIN1=(IH-1)*60+1
      MIN2=MIN1+59
      DO 32 IM=1,4
      IMS(IM)=0
      AMX(IM)=0.
      DO 31 I2=MIN1,MIN2
      IFLAG=0
      IM2=IM-1
      IF(IM2.EQ.0) GO TO 35
      DO 7 J=1,IM2
      IF(IMS(J).EQ.I2) IFLAG=1
    7 CONTINUE
      IF(IFLAG.EQ.1) GO TO 31
   35 CONTINUE
      AMX(IM)=AMAX1(AMX(IM),WM(I2))
      IF((AMX(IM)-WM(I2)).LT.1.E-04) IMS(IM)=I2
   31 CONTINUE
   32 CONTINUE
      FOUR=0.
      DO 33 J3=1,4
      I=IMS(J3)
      FOUR=FOUR+WM(I)
   33 CONTINUE
      TOT=0.
      DO 34 J3=MIN1,MIN2
   34 TOT=TOT+WM(J3)
      IF(TOT.LT.1.E-4) GO TO 30
      CONCEN(IH)=FOUR/TOT*100.
   30 CONTINUE
C
C       GET TOTAL DAILY VOLUME
C
      TOTAL=0.
      DO 9 JH=1,24
    9 TOTAL=TOTAL+WH(JH)
C
C           NORMALIZE MINUTE-BY-MINUTE ARRIVALS AND DEPARTURES
C
      DO 910 NORM1=1,1440
      DO 910 NORM2=1,2
      VOL(NORM1,NORM2)=VOL(NORM1,NORM2)/TOTAL
  910 CONTINUE
C
C            CALCULATE TODAY'S PEAKING FACTOR
C
      PF=0.
      DO 401 IP=1,23
      IP2=IP+1
      PF1=WH(IP)+WH(IP2)
      PF=AMAX1(PF,PF1)
  401 CONTINUE
      PF=PF/2./TOTAL
C
C         WRITE PROFILE INFORMATION TO DISK AND TAPE
C
      WRITE(IOUT) IDAY,WQ
C
      WRITE(ITAPE) IDAY,TOTAL,VOL,ADR,CONCEN,PMIXP,PF
C
C     PRINTING OPTION
C
      IF((.NOT.IPRINT).AND.(.NOT.IPRNT)) GO TO 904
      WRITE(3,901) AN(IA),YEAR,DATE(1),IDS,TOTAL,PF
  901 FORMAT(1H1/30X,A3,2X,I4,'/',I2,'/',I2/
     1  30X,'TOTAL VOLUME = 'F10.1,10X,' PEAKING FACTOR ='F12.4)
      DO 902 JH=1,24
      ARR=VOLH(JH,1)
      DEP=VOLH(JH,2)
      MIN1=(JH-1)*60+1
      MIN2=MIN1+59
      IQ1=(JH-1)*4+1
      IQ2=IQ1+3
      IF(IPRINT) WRITE(3,906) JH,WH(JH),ADR(JH),CONCEN(JH),
     1 (PMIXP(JH,J2),J2=1,8),
     2 (WQ(I2),I2=IQ1,1Q2),(WM(I1),I1=MIN1,MIN2)
      IF(IPRNT)  WRITE(3,906) JH,WH(JH),ADR(JH),CONCEN(JH),
     1(PMIXP(JH,J2),J2=1,8)
  906 FORMAT(I3,F6.0,' A/D RATIO ='F12.2,'  CONCENTRATION ='F10.4/
     1 10X,8F8.3/
     3 4F6.0/2(30F6.0/))
  902 CONTINUE
  904 CONTINUE
      GO TO 550
  505 CONTINUE
      GO TO 509
  507 WRITE(5,10)
   10 FORMAT('  YOU ARE SOMEWHAT MISTAKEN '
     1 /'       AS TO THE INFORMATION ON THIS TAPE'/)
  509 CONTINUE
      END FILE ITAPE
      END FILE IOUT
      WRITE(5,3305)
 3305 FORMAT(' DELETE FILE DIS??.DAT BEFORE LOGGING OUT ')
      GO TO 4712
 4711 REREAD 4713,(ANS(JRR),JRR=1,17)
 4713 FORMAT(1X,16A5,A3)
      WRITE(5,4713) (ANS(JRR),JRR=1,17)
      GO TO 502
 4712 CONTINUE
      END
```

APPENDIX B \

```
**********************************************************************
*                        CLUSTERING PROGRAMS
**********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:PCLUS.F4  [4255,516] 18-May-7
**********************************************************************
C           PCLUST.F4
C
      DIMENSION ARRAY(25000)
      DIMENSION MNTH(12)
      INTEGER AIR,YEAR
      COMMON/KDEV/KWHERE,MNTH,YEAR,NORM
      DATA ICL,IAND/"41630,"777777700000/
      DATA MNTH/31,28,31,30,31,30,31,31,30,31,30,31/
      LOGICAL NORM
      WRITE(5,101)
  101 FORMAT('  INPUT FILE NAME?'/)
      READ(5,102) NAME
  102 FORMAT(A5)
      CALL IFILE(20,NAME)
      WRITE(5,105)
  105 FORMAT('    DO YOU WANT TREES PRINTED?'/)
      KWHERE=5
      READ(5,205) REPLY
  205 FORMAT(A1)
      IF(REPLY.EQ.'N') KWHERE=0
  104 FORMAT(I)
      WRITE(5,106)
  106 FORMAT('  WHAT YEAR?'/)
      READ(5,104) YEAR
      IF(YEAR.LT.1900) YEAR=YEAR+1900
      IF(YEAR/4*4.EQ.YEAR) MNTH(2)=29
      YEAR=YEAR-1900
      WRITE(5,103)
  103 FORMAT('  HOW MANY AIRPORTS?'/)
      READ(5,104) NA
      WRITE(5,107)
  107 FORMAT('  DO YOU WANT NORMALIZED DATA?'/)
      READ(5,108) REPLY
  108 FORMAT(A1)
      NORM=.FALSE.
      IF(REPLY.EQ.'N') NORM=.TRUE.
      DO 55 JA=1,NA
      READ(20) AIR,NDAYS
      NAMEC=(AIR.AND.IAND).OR.ICL
      CALL OFILE(21,NAMEC)
C
C    WRITE TO 21 NC(=3),NDC(253),VOLD(253) = DAILY VOLUMES
C
C -         SUM QUARTER-HOURS TO GET DVOL(N) - IN CLSSUB
C
```

```
      CALL CLSSUB(AIR,ARRAY,NDAYS,96)
   55 CONTINUE
      END
```

```
**********************************************************************
*                         CLUSTERING PROGRAMS
**********************************************************************
      SUBROUTINE CLSSUB(AIR,ARRAY,NROWS,NCOLS)
      DIMENSION ARRAY(NROWS,NCOLS),IDAY(253),JJ(2),IDATE(2,2),
     1 NDC(253),DVOL(253),MNTH(12),NAME(253),THRES(253),IA(96)
      COMMON/KDEV/KWHERE,MNTH,IYEAR,NORM
      LOGICAL NORM
      INTEGER AIR
            DO 5 J=1,NROWS
      READ(20) IDAY(J),(ARRAY(J,K),K=1,NCOLS)
C      DO 1101 J2=1,NCOLS
C 1101 IA(J2)=ARRAY(J,J2)
C      WRITE(5,1102) IA
C 1102 FORMAT(24I3)
      IF(NORM) GO TO 5
C
C        NORMALIZE
C
      SUM=0.
      DO 15 JS=1,NCOLS
   15 SUM=SUM+ARRAY(J,JS)
      DO 16 JS=1,NCOLS
   16 ARRAY(J,JS)=ARRAY(J,JS)/SUM*100.
    5 CONTINUE
      JJ(1)=1
      JJ(2)=NROWS
      DO 50 J=1,2
      N=JJ(J)
      NDAY=IDAY(N)
      MONTH=1
   90 IF(NDAY.LE.MNTH(MONTH)) GO TO 80
      NDAY=NDAY-MNTH(MONTH)
      MONTH=MONTH+1
      GO TO 90
   80 CONTINUE
      IDATE(1,J)=MONTH
      IDATE(2,J)=NDAY
   50 CONTINUE
      IF(KWHERE) 66,66,65
   65 CONTINUE
      WRITE(KWHERE,60) AIR,IDATE(1,1),IDATE(2,1),IYEAR,
     1   IDATE(1,2),IDATE(2,2),IYEAR
   60 FORMAT(///10X,' AIRPORT  :   'A3/5X,I2,'/',I2,'/',I2,5X,'TO',
     1 5X,I2,'/',I2,'/',I2)
   66 CONTINUE
      CALL EUCLID(ARRAY,NROWS,NCOLS)
      CALL HICLUS(NAME,THRES,NROWS)
      IF(KWHERE) 76,76,75
   75 CONTINUE
```

```
      CALL PRTREE(NAME,THRES,THRES,NROWS,KWHERE)
   76 CONTINUE
      CALL YAWN(NAME,THRES,NDC,NROWS)
      DO 501 J11=1,NROWS
      DVOL(J11)=0.
      DO 499 J12=1,NCOLS
  499 DVOL(J11)=DVOL(J11)+ARRAY(J11,J12)
  501 CONTINUE
      NC=3
      WRITE(21) NC,NDC,DVOL
      RETURN
      END
```

```
**********************************************************************
*                         CLUSTERING PROGRAMS
**********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:PRFMRG.F4 [4255,516] 18-May-7
**********************************************************************
C                    PRFMRG.F4
C
C            THIS PROGRAM MERGES,FOR EACH OF
C            THIRTY-ONE AIRPORTS, EIGHT DISK FILES, EACH CONTAINING
C            QUARTER-HOUR PROFILES FOR ONE MONTH, PREPARATORY
C            TO CLUSTERING.   THIS FILES WERE PRODUCED BY OAGPRF.
C
      DIMENSION ARRAY(96),AN(8),ND(8),NCD(8,2)
      DIMENSION ITAG(8)
      DATA IAND,IPM,ITAG/"777777700000,"50232,"31142,"31144,
     1 "31146,"31150,"31542,"31544,"31546,"31550/
      INTEGER AN,ANS(31)
      DATA ANS/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
      WRITE(5,104)
  104 FORMAT('   TYPE IN LOCATIONS IN LIST OF',
     1' FIRST AND LAST AIRPORTS WANTED'/)
      READ(5,114) NA1,NA2
  114 FORMAT(2I)
  102 FORMAT(A5)
      WRITE(5,204)
  204 FORMAT('  NUMBERS OF FIRST AND LAST FILES TO BE MERGED?'/)
      READ(5,114) NF1,NF2
      NF=8
  105 FORMAT(I)
      DO 100 IA=NA1,NA2
      NAME1=ANS(IA)
      DO 10 I=NF1,NF2
      NAME=(NAME1.AND.IAND).OR.ITAG(I)
      IDEV=I+15
      CALL ASSDEV(IDEV,'DSK')
      CALL IFILE(IDEV,NAME)
   10 CONTINUE
      NAMEO=(NAME1.AND.IAND).OR.IPM
      CALL OFILE(24,NAMEO)
C
C          READ AIRPORT AND NUMBER OF DAYS
C           FROM EACH INPUT FILE
C
      DO 20 I=NF1,NF2
      IDEV=I+15
      READ(IDEV) AN(I),ND(I)
      WRITE(5,2020) AN(I),ND(I)
```

```
2020 FORMAT(5X,A3,I5)
  20 CONTINUE
C
C     CHECK THAT AIRPORT IS THE SAME ON ALL FILES
C
      NF3=NF1+1
      DO 30 I=NF3,NF2
      N2=I-1
      IF(AN(N2).NE.AN(NF2)) GO TO 150
  30 CONTINUE
C
C          SUM DAYS FROM INPUT FILES
C
      NDAYS=0
      DO 473 IADD=NF1,NF2
 473 NDAYS=NDAYS+ND(IADD)
      WRITE(24) AN(NF1),NDAYS
      DO 40 I=NF1,NF2
      IDEV=15+I
      JD=ND(1)
      DO 41 J=1,JD
      READ(IDEV) IDAY,ARRAY
      WRITE(24) IDAY,ARRAY
      IF(J.EQ.1) NCD(I,J)=IDAY
      IF(J.EQ.JD) NCD(I,2)=IDAY
  41 CONTINUE
  40 CONTINUE
      WRITE(5,202) AN(NF1)
 202 FORMAT('  AIRPORT= 'A3)
      WRITE(5,201) ((NCD(I,J),J=1,2),I=NF1,NF2)
 201 FORMAT('  DAY OF YEAR ',I5,'  TO ',I5)
      GO TO 250
 150 WRITE(5,155) AN
 155 FORMAT('  AIRPORTS INCONSISTENT '/20X,3(A3,5X))
 250 CONTINUE
      END FILE 24
      DO 107 I=NF1,NF2
      IDEV=I+15
      CALL RELEAS(IDEV)
 107 CONTINUE
      CALL RELEAS(24)
      WRITE(5,127) ANS(IA)
 127 FORMAT('   AIRPORT:'A3,1X,'DONE')
 100 CONTINUE
      END
```

```
**********************************************************************
*                         CLUSTERING PROGRAMS
**********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:CLUSTR.F4 [4255,516] 18-May-7(
**********************************************************************
C             CLUSTR.F4
C
C             THIS PROGRAM CLUSTERS QUARTER-HOUR PROFILES
C             READ FROM DISK FILES WRITTEN BY PRFMRG.
C
      DIMENSION ARRAY(12000)
      INTEGER AIR
      COMMON/KDEV/KWHERE,NORM
      DATA ICL,IAND/"41630,"777777700000/
      COMMON /CLUS/NC,NOTREE
      COMMON /AIRPL/AIR
      LOGICAL NORM,NOTREE
      INTEGER AIR1,AN(31)
      DATA AN/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
      DATA IPM/"50232/
      WRITE(5,105)
  105 FORMAT('   DO YOU WANT TREES PRINTED?'/)
      KWHERE=5
      READ(5,205) REPLY
  205 FORMAT(A1)
      IF(REPLY.EQ.'N') KWHERE=0
  104 FORMAT(2I)
      WRITE(5,100)
  100 FORMAT('  HOW MANY CLUSTERS?'/)
      READ(5,104) NC
      WRITE(5,102)
  102 FORMAT('  SHALL I SAVE THE TREES?'/)
      READ(5,108) REPLY
      NOTREE=.TRUE.
      IF(REPLY.EQ.'Y') NOTREE=.FALSE.
      WRITE(5,103)
  103 FORMAT('  LOCATIONS IN LIST OF FIRST AND LAST AIRPORT?'/)
      READ(5,104) NA1,NA2
      WRITE(5,107)
  107 FORMAT('  DO YOU WANT NORMALIZED DATA?'/)
      READ(5,108) REPLY
  108 FORMAT(A1)
      NORM=.FALSE.
      IF(REPLY.EQ.'N') NORM=.TRUE.
      DO 55 JA=NA1,NA2
      AIR1=AN(JA)
      NAME1=(AIR1.AND.IAND).OR.IPM
      CALL IFILE(20,NAME1)
        READ(20) AIR,NDAYS
      NAMEC=(AIR.AND.IAND).OR.ICL
      CALL OFILE(21,NAMEC)
C
      CALL CLSSUB(AIR,ARRAY,NDAYS,96)
      CALL RELEAS(20)
   55 CONTINUE
      END
```

```
*******************************************************************************
*                            CLUSTERING PROGRAMS
*******************************************************************************
      SUBROUTINE CLSSUB(AIR,ARRAY,NROWS,NCOLS)
      DIMENSION ARRAY(NROWS,NCOLS),IDAY(120),
     1 NDC(120),DVOL(120),NAME(120),THRES(120),IA(96)
      COMMON/KDEV/KWHERE,NORM
      COMMON/CLUS/NC
      LOGICAL NORM
      INTEGER AIR
             DO 5 J=1,NROWS
      READ(20) IDAY(J),(ARRAY(J,K),K=1,NCOLS)
C
C         NORMALIZE
C
      SUM=0.
      DO 15 JS=1,NCOLS
   15 SUM=SUM+ARRAY(J,JS)
      DVOL(J)=SUM
      IF(NORM) GO TO 5
      DO 16 JS=1,NCOLS
   16 ARRAY(J,JS)=ARRAY(J,JS)/SUM*100.
    5 CONTINUE
      IF(KWHERE) 66,66,65
   65 CONTINUE
      WRITE(KWHERE,60) AIR
   60 FORMAT(///10X,' AIRPORT  :    'A3//)
   66 CONTINUE
      CALL EUCLID(ARRAY,NROWS,NCOLS)
      CALL HICLUS(NAME,THRES,NROWS)
      IF(KWHERE) 76,76,75
   75 CONTINUE
      CALL PRTREE(NAME,THRES,THRES,NROWS,KWHERE)
   76 CONTINUE
      CALL SUSPCL(NAME,THRES,NDC,NROWS)
      WRITE(21) NC,NDC,DVOL
      RETURN
      END
      SUBROUTINE EUCLID(AMAT,NROW,NCOL)
      DIMENSION AMAT(NROW,NCOL)
      COMMON /CLUST/ PROX(8000)
      INDEX=0
      DO 30 I=2,NROW
      II=I-1
      DO 20 J=1,II
      DIST=0
      INDEX=INDEX+1
      DO 10 K=1,NCOL
      DIST=DIST+(AMAT(I,K)-AMAT(J,K))**2
 10   CONTINUE
      PROX(INDEX)=SQRT(DIST)
 20   CONTINUE
 30   CONTINUE
      RETURN
      END
```

```
************************************************************************************
*                              CLUSTERING PROGRAMS
************************************************************************************
      SUBROUTINE HICLUS(NAME,THRES,NROWS)
      DIMENSION  TREE(120,3),LINE(120)
      DIMENSION LAMBDA(120),THRES(NROWS),NAME(NROWS),OUTPUT(120,3)
      COMMON /CLUST/ PROX(8000)
      IPLACE(K)=(3.+SQRT(9.-8.*(2.-FLOAT(K))))/2.
      JPLACE(K)=K-(IPLACE(K)-1)*(IPLACE(K)-2)/2
       DO 483 JJJ=1,NROWS
  483 LINE(JJJ)=0
      N=NROWS-1
      KOUNT=1
      LITTLE=1.
      BIG=0.
      NUMBER=NROWS*(NROWS-1)/2

C     DETERMINE MAXIMAL ELEMENT OF MATRIX (BIG)

      DO 1 KHL=1,NUMBER
       IF(BIG.LT.PROX(KHL)) BIG=PROX(KHL)
 1    CONTINUE
       BIG=2.*BIG

C     DETERMINE PRESENT MINIMAL ELEMENT OF MATRIX

 2    DO 3 LOO=1,NUMBER
      IF (PROX(LITTLE) .LT. PROX(LOO)) GOTO 3
      LITTLE=LOO
 3    CONTINUE
      TREE(KOUNT,1)=PROX(LITTLE)
      TREE(KOUNT,2)=IPLACE(LITTLE)
      TREE(KOUNT,3)=JPLACE(LITTLE)
      IF(KOUNT .EQ.NROWS-1) GOTO 5
      PROX(LITTLE)=BIG
      KOUNT=KOUNT+1
C
C     DERIVE MEAN OF CLUSTERED ROWS
C
      DO 4 K2=1,NROWS
      IF(IPLACE(LITTLE).EQ.K2) GOTO 4
      IF(JPLACE(LITTLE).EQ.K2) GOTO 4
      IK=ISQ2TR(IPLACE(LITTLE),K2)
      JK=ISQ2TR(JPLACE(LITTLE),K2)
      PROX(IK)=(PROX(IK)+PROX(JK))/2.
      PROX(JK)=BIG
 4    CONTINUE
      GOTO 2

C     INITIATE SORTING PARAMETERS
```

AD-A062 881 TRANSPORTATION SYSTEMS CENTER CAMBRIDGE MASS F/G 1/5
THE AIRPORT PERFORMANCE MODEL. VOLUME II. USER'S MANUAL AND PRO--ETC(U)
OCT 78 J BELLANTONI
UNCLASSIFIED TSC-FAA-28-21.2 FAA-ASP-78-10.2 NL

2 OF 3
ADA
062881

```
 5     LEVEL=1
       LAMBDA(1)=N
       THRES(1)=TREE(1,2)
       I=1
       K=2

C      BEGIN SORTING PROCESS

       DO 6 M6=1,3
       OUTPUT(1,M6)=TREE(1,M6)
       TREE(1,M6)=0.
 6     CONTINUE
 10    IF(TREE(I,2).NE.THRES(LEVEL)) GOTO 20
       LEVEL=LEVEL+1
       THRES(LEVEL)=TREE(I,3)
       LAMBDA(LEVEL)=I
       DO 11 M11=1,3
       OUTPUT(K,M11)=TREE(I,M11)
       TREE(I,M11)=0.
 11    CONTINUE
       IF(K.EQ.N) GOTO 60
       K=K+1
       I=1
       GOTO 10
 20    IF(I.LT.LAMBDA(LEVEL)) GOTO 30
       IF(LEVEL.EQ.1) GOTO 50
       LEVEL=LEVEL-1
       GOTO 10
 30    IF(LEVEL.NE.1) GOTO 40
       IF(TREE(I,3).EQ.THRES(LEVEL)) GOTO 50
 40    I=I+1
       GOTO 10
50     THRES(LEVEL)=TREE(I,2)
       DO 51 M51=1,3
       OUTPUT(K,M51)=TREE(I,M51)
       TREE(I,M51)=0.
 51    CONTINUE
       IF(K.EQ.N) GOTO 60
       K=K+1
       I=1
       GOTO 10

C      STORE SORTED  THRESHOLD VALUES IN THRES

 60     DO 70 M70=1,N
       THRES(M70)=OUTPUT(M70,1)
 70  CONTINUE

C    SORT CELL NUMBERS FOR PRINTING

186  NAME(1)=OUTPUT(1,2)
     NN=NAME(1)
     LINE(NN)=1
     NAME(2)=OUTPUT(1,3)
     NN=NAME(2)
     LINE(NN)=1
     DO 200 IOTA=3,NROWS
     NN=OUTPUT(IOTA-1,2)
     IF(LINE(NN).EQ.1) GOTO 190
     NAME(IOTA)=NN
     LINE(NN)=1
     GOTO 200
190  NAME(IOTA)=OUTPUT(IOTA-1,3)
     NN=NAME(IOTA)
     LINE(NN)=1
200  CONTINUE
     RETURN
     END
```

```
*******************************************************************************
*                             CLUSTERING PROGRAMS
*******************************************************************************
      FUNCTION ISQ2TR(I,J)
      IF(I.GT.J) GOTO 10
      ISQ2TR=((J-1)*(J-2)/2)+I
      IF(I.EQ.J) ISQ2TR=0
      RETURN
10    ISQ2TR=((I-1)*(I-2)/2)+J
       RETURN
       END
      SUBROUTINE PRTREE(NAME,TAG,PRTLST,M,KWHERE)
      DIMENSION TAG(M), MU(120),MARK(120),NAME(M),LAMBDA(120)
      DIMENSION LETTER(150),IMARK(150),INTER(3),LICE(3)
      DIMENSION PRTLST(M)
      LOGICAL FLAG
      DATA KBLANK /' '/
      DATA KDASH /'-'/
      DATA KQUOTE /''''/
      DATA KPOINT /'.'/
      DATA KI /'I'/
       ICE=0
       KRYPTO=0
       DO 445 JQ=1,M
       MARK(JQ)=0
       MU(JQ)=0
       LAMBDA(JQ)=0
  445  CONTINUE
       DO 545 JQ=1,150
       IMARK(JQ)=0
       LETTER(JQ)=0
  545  CONTINUE
      N=M-1
C
C     CHECK WHETHER OUTPUT IS TO BE ON-LINE OR OFF-LINE
C
      IF(KWHERE.EQ.3) GOTO 1234
      DEVICE=59.
      KDEV1=64
      KDEV2=65
      GOTO 5678
 1234        DEVICE=99.
      KDEV1=104
      KDEV2=105
 5678        BIG=0
      DO 70 M70=1,N
      IF(TAG(M70).LT.BIG) GOTO 70
      BIG=TAG(M70)
 70   CONTINUE
C
C     NORMALIZE BOUNDS FOR PRINTING ON TTY
C
      SMALL=BIG
      DO 80 M80=1,N
      IF(TAG(M80).GT.SMALL) GOTO 80
      SMALL=TAG(M80)
 80   CONTINUE
```

```
      DO 90 M90=1,N
      LAMBDA(M90)=4.+(TAG(M90)-SMALL)*DEVICE/(BIG-SMALL)
 90   CONTINUE
C
C     DETERMINE HOW FAR TO THE RIGHT EACH LINE EXTENDS
C
      IFIRST=0
      LARGE=0
      I=1
 100  IF(MARK(I).NE.0) GOTO 110
      IF(LARGE.GT.LAMBDA(I)) GOTO 130
      MOON=I
      LARGE=LAMBDA(I)
      GOTO 130
 110  IF(MARK(I).EQ.KRYPTO) GOTO 120
      IF(LARGE.EQ.0) GOTO 130
      MU(MOON)=LAMBDA(JUNE)
      ICE=ICE+1
      MARK(MOON)=KRYPTO+1
      LARGE=0
      GOTO 130
 120  JUNE=I
      IF(LARGE.EQ.0) GOTO 130
      MU(MOON)=LAMBDA(I)
      ICE=ICE+1
      LARGE=0
      MARK(MOON)=KRYPTO+1
 130  IF(I.EQ.N) GOTO 140
      I=I+1
      GOTO 100
 140  IF(ICE.EQ.N) GOTO 220
      IF(IFIRST.EQ.1) GOTO 150
      MU(MOON)=KDEV1
      ICE=ICE+1
      MARK(MOON)=1
      IFIRST=1
      GOTO 160
 150  IF(LARGE.EQ.0) GOTO 160
      MU(MOON)=LAMBDA(JUNE)
      ICF=ICE+1
      MARK(MOON)=KRYPTO+1
 160  KRYPTO=KRYPTO+1
      I=1
      LARGE=0
      GOTO 100

220   DO 230 M230=2,N
```

```
 225 MARK(M230)=1
     IF(LAMBDA(M230).GT.LAMBDA(M230-1)) GOTO 230
     MARK(M230)=0
 230 CONTINUE
     MARK(1)=0

C    BIG LOOP STARTS HERE

     DO 500 MOO=1,N
     FLAG=.FALSE.

C    BLANK OUT THE LINE

 260 DO 270 M270=1,KDEV2
 270 LETTER(M270)=KBLANK
     IF(FLAG) GOTO 290

     LICE(1)=NAME(MOO)/100
     LICE(2)=(NAME(MOO)-LICE(1)*100)/10
     LICE(3)=NAME(MOO)-LICE(1)*100-LICE(2)*10
     ENCODE(1,1003,INTER(1)) LICE(1)
     ENCODE(1,1003,INTER(2)) LICE(2)
     ENCODE(1,1003,INTER(3)) LICE(3)
     IF(MARK(MOO).EQ.1) GOTO 280
     LL=LAMBDA(MOO)-3
     LETTER(LL)=INTER(1)
     LL=LL+1
     LETTER(LL)=INTER(2)
     LL=LL+1
     LETTER(LL)=INTER(3)
     LL=LL+1
     LETTER(LL)=KDASH
     LL=LL+1

     LETTER(LL)=KPOINT
     IF(LAMBDA(MOO).EQ.LAMBDA(MOO-1)) LETTER(LL)=KI
     IMARK(LL)=1
     GOTO 320

 280 NARK=1
     DO 285 M285=1,KDEV1
     IF(NARK.EQ.0) GOTO 285
     IF(IMARK(M285).EQ.0) GOTO 285
     IMARK(M285)=0
     NARK=0
     LL=M285-4
```

```
285 CONTINUE
    LETTER(LL)=INTER(1)
    LL=LL+1
    LETTER(LL)=INTER(2)
    LL=LL+1
    LETTER(LL)=INTER(3)
    LL=LL+1
    LETTER(LL)=KDASH
    LL=LL+1
    LETTER(LL)=KQUOTE
    IF(LAMBDA(MOO).LT.MU(MOO)) GOTO 320
    LETTER(LL)=KI
    IMARK(LL)=1
    GOTO 320
    GOTO 320

290 LL=LAMBDA(MOO)+1
    IMARK(LL)=1
    IF( LAMBDA(MOO)+2.GT.MU(MOO)) GOTO 301
295 DO 300 M300=LAMBDA(MOO)+2,MU(MOO)
300 LETTER(M300)=KDASH

301 LL=MU(MOO)+1
    IF(LL.NE.KDEV2) GOTO 305
    IMARK(KDEV1)=1
    LETTER(KDEV2)=KDASH
    GOTO 320

305 IF(IMARK(LL).EQ.0) GOTO 310
    LETTER(LL)=KQUOTE
    IF(MU(MOO).EQ.LAMBDA(MOO+1)) GOTO 3055
    IF(LAMBDA(MOO).LT.MU(MOO)) GOTO 306
3055          LETTER(LL)=KI
    GOTO 320
306 IMARK(LL)=0
    GOTO 320
310 LETTER(LL)=KPOINT
    IMARK(LL)=1
320 DO 340 M340=1,KDEV1
    IF(IMARK(M340).EQ.0) GOTO 340
    IF(M340.EQ.LL) GOTO 340
    LETTER(M340)=KI
340 CONTINUE

350 IF(FLAG) GOTO 360
    WRITE(KWHERE,1000) (LETTER(KOW),KOW=1,KDEV2)
355 FLAG=.TRUE.
```

```
    GOTO 260

360 WRITE(KWHERE,1001) PRTLST(MOO) ,(LETTER(KOW) ,KOW=1,KDEV2)
 500        CONTINUE

    DO 370 M370=1,KDEV2
370 LETTER(M370)=KBLANK
    LICE(1)=NAME(M)/100
    LICE(2)=(NAME(M)-LICE(1)*100)/10
    LICE(3)=NAME(M)-LICE(1)*100-LICE(2)*10
    ENCODE(1,1003,INTER(1)) LICE(1)
    ENCODE(1,1003,INTER(2)) LICE(2)
    ENCODE(1,1003,INTER(3)) LICE(3)

    LL=LAMBDA(N)-3
    LETTER(LL)=INTER(1)
    LL=LL+1
    LETTER(LL)=INTER(2)
    LL=LL+1
    LETTER(LL)=INTER(3)
    LL=LL+1
    LETTER(LL)=KDASH
    LL=LL+1
    LETTER(LL)=KQUOTE
    WRITE(KWHERE,1000) (LETTER(KOW),KOW=1,KDEV2)
    RETURN
1000        FORMAT(T9,104A1)
1001        FORMAT(F9.3,T9,104A1)
1003        FORMAT(I1)
     RETURN
     END
```

```
**************************************************************************
*                                CLUSTERING PROGRAMS
**************************************************************************
      SUBROUTINE YAWN(NAME,THRES,NDC,NUM)
      DIMENSION NDC(NUM),NAME(NUM),THRES(NUM),INDEX(2)
C
C              NAME IS THE ARRAY OF DAY NUMBERS
C              THRES IS THE ARRAY OF (SURPRISE!) THRESHOLD
C                 VALUES
C              INDEX IS WHAT YOU GET OUT
C              NUM IS 250-WHATEVER
C              GOOD LUCK
C
C              NDC(I) ASSIGNS A CLUSTER NUMBER TO DAY I
C
      LENIN=0
      KGB=0
      CCCP=0.
    1 DO 10 MARX=1,NUM-1
      IF(THRES(MARX).LT.CCCP) GO TO 10
      IF(MARX.EQ.LENIN) GO TO 10
      CCCP=THRES(MARX)
      KGB=MARX
   10 CONTINUE
      IF(LENIN.NE.0) GO TO 20
      LENIN=KGB
      INDEX(1)=KGB
      CCCP=0.
      GO TO 1
   20 INDEX(2)=KGB
      I1=INDEX(1)
      I2=INDEX(2)
      IF(I1.LT.I2) GO TO 40
       I2=I1
      I1=INDEX(2)
   40 CONTINUE
      DO 28 I=1,I1
      J=NAME(I)
   28 NDC(J)=1
      DO 29 I=I1+1,I2
      J=NAME(I)
   29 NDC(J)=2
      DO 30 I=I2+1,NUM
      J=NAME(I)
   30 NDC(J)=3
      RETURN
      END
```

```
*****************************************************************************
*                             CLUSTERING PROGRAMS
*****************************************************************************
      SUBROUTINE SUSPCL(NAME,THRES,NDC,NUM)
      LOGICAL NOTREE
      COMMON/AIRPL/AIR
      COMMON /CLUS/NC,NOTREE
      INTEGER AIR
      DATA IST,IAND/"51650,"777777700000/
      DIMENSION IORD(120)
      DIMENSION NDC(NUM),NAME(NUM),THRES(NUM),INDEX(0/10)
      IF(NOTREE) GO TO 250
      CALL ASSDEV(16,'DSK')
      NAMEF=IST.OR.(AIR.AND.IAND)
       CALL OFILE(16,NAMEF)
      WRITE(16) NUM
      WRITE(16) NAME,THRES
      CALL RELEAS(16)
 250  CONTINUE
      IF(NC.EQ.3) GO TO 7000
      NTH=NUM-1
      DO 200 IN=1,NTH
 200  IORD(IN)=IN
      I=0
      ISAVE=0
 800  I=ISAVE+1
      ISAVE=I
      I=I-1
 600  I=I+1
      IF(I.LT.1) GO TO 800
      IF(I+1.GT.NTH) GO TO 500
       I1=IORD(I)
      I2=IORD(I+1)
      IF(THRES(I1).LE.THRES(I2)) GO TO 800
      IORD(I+1)=I1
      IORD(I)=I2
      I=I-2
      GO TO 600
 500  CONTINUE
      NBREAK=NC-1
      N=NTH-NBREAK+1
      DO 201 IBRK=1,NBREAK
      INDEX(IBRK)=IORD(N)
      N=N+1
 201  CONTINUE
      I=0
      ISAVE=0
 880  I=ISAVE+1
      ISAVE=I
      I=I-1
```

```
  660 I=I+1
      IF(I.LT.1) GO TO 880
      IF(I+1.GT.NBREAK) GO TO 550
      I1=INDEX(I)
      I2=INDEX(I+1)
      IF(I1.LE.I2) GO TO 880
      INDEX(I+1)=I1
      INDEX(I)=I2
      I=I-2
      GO TO 660
  550 CONTINUE
      INDEX(0)=0
      INDEX(NC)=NUM
      DO 202 I=1,NC
      LIMIT1=INDEX(I-1)+1
      LIMIT2=INDEX(I)
      DO 202 LIM=LIMIT1,LIMIT2
      J=NAME(LIM)
      NDC(J)=I
  202 CONTINUE
      GO TO 7001
 7000 CALL YAWN(NAME,THRES,NDC,NUM)
 7001 CONTINUE
      RETURN
      END
```

```
*********************************************************************************
*                           CLUSTERING PROGRAMS
*********************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:PRFPLT.F4 [4255,516] 18-May-7
*********************************************************************************
C               PRFPLT.F4
C
        DIMENSION ARRAY(25000)
        DIMENSION MNTH(12)
        INTEGER YEAR
        COMMON/KDEV/KWHERE,MNTH,YEAR,NORM
        DATA MNTH/31,28,31,30,31,30,31,31,30,31,30,31/
        LOGICAL NORM
        WRITE(5,101)
    101 FORMAT('  INPUT FILE NAME?'/)
        READ(5,102) NAME
    102 FORMAT(A5)
        CALL IFILE(20,NAME)
    104 FORMAT(I)
        WRITE(5,106)
    106 FORMAT('  WHAT YEAR?'/)
        READ(5,104) YEAR
        IF(YEAR.LT.1900) YEAR=YEAR+1900
        IF(YEAR/4*4.EQ.YEAR) MNTH(2)=29
        YEAR=YEAR-1900
        WRITE(5,103)
    103 FORMAT('  HOW MANY AIRPORTS?'/)
        READ(5,104) NA
        WRITE(5,107)
    107 FORMAT('  DO YOU WANT NORMALIZED DATA?'/)
        READ(5,108) REPLY
    108 FORMAT(A1)
        NORM=.FALSE.
        IF(REPLY.EQ.'N') NORM=.TRUE.
        DO 55 JA=1,NA
        READ(20) AIR,NDAYS
        CALL PLTSUB(AIR,ARRAY,NDAYS,96)
     55 CONTINUE
        END
```

```
**********************************************************************
*                         CLUSTERING PROGRAMS
**********************************************************************
      SUBROUTINE PLTSUB(AIR,ARRAY,NROWS,NCOLS)
      DIMENSION II1(3),II2(3)
      DATA II1,II2/1,30,92,29,91,241/
      DIMENSION ARRAY(NROWS,NCOLS),IDAY(241),JJ(2),IDATE(2,2),
     1 MNTH(12),NAME(241),THRES(241),IA(96),
     2 IPLT(10),YPLT(96),NDJA(241)
      COMMON/KDEV/KWHERE,MNTH,IYEAR,NORM
      LOGICAL NORM
            DO 5 J=1,NROWS
      READ(20) IDAY(J),(ARRAY(J,K),K=1,NCOLS)
C      DO 1101 J2=1,NCOLS
C 1101 IA(J2)=ARRAY(J,J2)
C      WRITE(5,1102) IA
C 1102 FORMAT(24I3)
      IF(NORM) GO TO 5
C
C        NORMALIZE
C
      SUM=0.
      DO 15 JS=1,NCOLS
   15 SUM=SUM+ARRAY(J,JS)
      DO 16 JS=1,NCOLS
   16 ARRAY(J,JS)=ARRAY(J,JS)/SUM*100.
    5 CONTINUE
      JJ(1)=1
      JJ(2)=NROWS
      DO 50 J=1,2
      N=JJ(J)
      NDAY=IDAY(N)
      MONTH=1
   90 IF(NDAY.LE.MNTH(MONTH)) GO TO 80
      NDAY=NDAY-MNTH(MONTH)
      MONTH=MONTH+1
      GO TO 90
   80 CONTINUE
      IDATE(1,J)=MONTH
      IDATE(2,J)=NDAY
   50 CONTINUE
      WRITE(5,60) AIR,IDATE(1,1),IDATE(2,1),IYEAR,
     1   IDATE(1,2),IDATE(2,2),IYEAR
   60 FORMAT(///10X,' AIRPORT  :   'A3/5X,I2,'/',I2,'/',I2,5X,'TO',
     1 5X,I2,'/',I2,'/',I2)
      WRITE(5,1407)
 1407 FORMAT('  WILL THIS RUN AVERAGE FOR CLUSTERS?'/)
      READ(5,1408) THIS
 1408 FORMAT(A1)
      IF(THIS.EQ.'Y') GO TO 1410
      WRITE(5,1402)
 1402 FORMAT(10X'  HOW MANY DAYS DO YOU WANT PLOTTED?'/)
      READ(5,1403) NPLT
 1403 FORMAT(10I)
      WRITE(5,1404)
 1404 FORMAT('  WHICH ARE THEY?'/)
      READ(5,1403) (IPLT(K),K=1,NPLT)
      DO 1405 JP=1,NPLT
      LPLT=IPLT(JP)
      DO 1406 J7=1,NCOLS
      YPLT(J7)=ARRAY(LPLT,J7)
 1406 CONTINUE
      CALL XYPLOT(1.,96.,1.,YPLT)
 1405 CONTINUE
      RETURN
 1410 CONTINUE
      WRITE(5,1411)
 1411 FORMAT('  HOW MANY CLUSTERS?'/)
      READ(5,1412) NCLUS
 1412 FORMAT(5I)
      WRITE(5,927)
  927 FORMAT('  DO YOU WANT TO USE CLUSTER DEFAULT?'/)
      READ(5,928) DEF
  928 FORMAT(A1)
      DO 1420 JCLUS=1,NCLUS
      IF(DEF.EQ.'Y') GO TO 929
      WRITE(5,1421) JCLUS
 1421 FORMAT('  HOW MANY DAYS IN CLUSTER'I4,'?'/)
      READ(5,1412) NDJ
      WRITE(5,1422)
 1422 FORMAT('  WHICH DAYS ARE THEY? 5I '/)
      READ(5,1412) (NDJA(II),II=1,NDJ)
      GO TO 930
  929 NDJ=II2(JCLUS)-II1(JCLUS)
      ILL=0
      DO 922 IGR=II1(JCLUS),II2(JCLUS)
      ILL=ILL+1
      NDJA(ILL)=IGR
  922 CONTINUE
  930 CONTINUE
      VSUM=0.
      DO 1425 IY=1,96
      YPLT(IY)=0.
      DO 1428 IY2=1,NDJ
      NJ=NDJA(IY2)
      YPLT(IY)=YPLT(IY)+ARRAY(NJ,IY)
      VSUM=VSUM+ARRAY(NJ,IY)
 1428 CONTINUE
      YPLT(IY)=YPLT(IY)/FLOAT(NDJ)
 1425 CONTINUE
      CALL XYPLOT(1.,96.,1.,YPLT)
      IF(.NOT.NORM) GO TO 1826
      VSUM=VSUM/FLOAT(NDJ)
      WRITE(5,1825) VSUM
 1825 FORMAT(///5X'FOR THIS CLUSTER, AVERAGE DAILY VOLUME IS',
     1 F12.2///)
 1826 CONTINUE
 1420 CONTINUE
      RETURN
      END
      SUBROUTINE XYPLOT(XMIN,XMAX,XDEL,Y)
      DIMENSION Y(1),ALPHA(61),FORM(5),TITLE(70)
      DATA (FORM(J),J=1,5)/'//(',' ','X,',' ','A1)'/
      WRITE(5,101)
  101 FORMAT(/1X,'HOW MANY CHARACTERS IN TITLE?'/)
      READ (5,100) NCH
  100         FORMAT(I)
      NBL=(72-NCH)/2
      ENCODE(5,203,FORM(2)) NBL
      ENCODE(5,203,FORM(4)) NCH
      WRITE(5,205)
  205 FORMAT(1X,'TYPE IN TITLE '/)
      READ(5,204) (TITLE(I),I=1,NCH)
  203 FORMAT(I5)
  204 FORMAT(70A1)
      WRITE(5,FORM) (TITLE(K),K=1,NCH)
      NSTARS=61
      MID=31
      YMIN=Y(1)
      YMAX=Y(1)
      X=XMIN+XDEL
      J=2
  400 YMIN=AMIN1(YMIN,Y(J))
      YMAX=AMAX1(YMAX,Y(J))
      X=XDEL+X
      J=J+1
      IF(X.LE.XMAX) GO TO 400
      YMID=(YMIN+YMAX)/2.0
      WRITE(5,52) YMIN,YMID,YMAX
   52 FORMAT(//10X,G12.5,13X,G12.5,12X,G12.5/)
      YINC=(YMAX-YMIN)/60.
      X=XMIN
      J=1
  300 DO 201 I=2,60
  201         ALPHA(I)=' '
      ALPHA(1)='.'
      ALPHA(MID)='.'
      ALPHA(61)='.'
      XNSTAR=(Y(J)-YMIN)/YINC
      NSTAR=IFIX(XNSTAR)+1
      IF((XNSTAR-NSTAR).GE.0.50) NSTAR=NSTAR+1
      DO 202 I=1,NSTAR
  202 ALPHA(I)='*'
      WRITE(5,50) X,ALPHA
   50         FORMAT(G9.3,1X,61A1)
      X=X+XDEL
      J=J+1
      IF(X.LE.XMAX) GO TO 300
      RETURN
      END
```

```
*****************************************************************************
*                        CLUSTERING PROGRAMS
*****************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:CORAL.F4   [4255,516]  18-May-7
*****************************************************************************
C              CORAL.F4
C
          DIMENSION ARRAY(9000),ARRAY2(3000)
      104 FORMAT(I)
          CALL IFILE(22,'FEB75')
          CALL IFILE(21,'ALL75')
          CALL IFILE(20,'CORAL')
                NA=5
          DO 55 JA=1,NA
          READ(21) AIR,NDAYS
          READ(22) AIR,NDAY2
          PEAD(20,100) IAIR
      100 FORMAT(A3)
          WRITE(3,101) IAIR
      101 FORMAT(1H1///20X,A3//)
          CALL CORSUB(JA,ARRAY,ARRAY2,NDAYS,NDAY2,96)
       55 CONTINUE
          END
      SUBROUTINE CORSUB(JA,ARRAY,ARRAY2,NROWS,NROWS2,NCOLS)
      DIMENSION NDJ(5),NDJ2(5)
      COMMON NCOR,DCOR,NLIST,DLIST
      INTEGER DLIST(5,4,2),DCOR(5)
      COMMON /SAVE/NCLUS,NCLUS2,NDJ,NDJ2
      DIMENSION NCOR(5),NLIST(5,9,2)
      DIMENSION X(96),Y(96)
      DIMENSION ARRAY(NROWS,NCOLS),
     1 ARRAY2(NROWS2,NCOLS),NAME(124),THRES(124),IA(96),
     A YPLT2(96,5),NDJA2(30,5),
     2 IPLT(10),YPLT(96,5),NDJA(90,5)
      DIMENSION IDAY(90)
             DO 5 J=1,NROWS
      READ(21) IDAY(J),(ARRAY(J,K),K=1,NCOLS)
      IF(J.LE.NROWS2) READ(22) ID726,(ARRAY2(J,K),K=1,NCOLS)
C
C           NORMALIZE
C
      SUM=0.
      DO 15 JS=1,NCOLS
   15 SUM=SUM+ARRAY(J,JS)
      DO 16 JS=1,NCOLS
   16 ARRAY(J,JS)=ARRAY(J,JS)/SUM
      IF(J.GT.NROWS2) GO TO 5
      SUM=0.
      DO 215 JS=1,NCOLS
  215 SUM=SUM+ARRAY2(J,JS)
      DO 216 JS=1,NCOLS
  216 ARRAY2(J,JS)=ARRAY2(J,JS)/SUM
    5 CONTINUE
      NST=NCLUS
      READ(20,1413) NCLUS
 1413 FORMAT(A4)
      IF(NCLUS.EQ.'SAME') GO TO 1423
      REREAD 1412,NCLUS
 1412 FORMAT(20I)
      DO 1422 JCLUS=1,NCLUS
      READ(20,1412) NDJ(JCLUS)
      ND=NDJ(JCLUS)
      READ(20,1412) (NDJA(II,JCLUS),II=1,ND)
 1422 CONTINUE
      GO TO 1424
 1423 NCLUS=NST
 1424 DO 1420 JCLUS=1,NCLUS
      DO 1425 IY=1,96
      YPLT(IY,JCLUS)=0.
      ND=NDJ(JCLUS)
      DO 1428 IY2=1,ND
      NJ=NDJA(IY2,JCLUS)
      YPLT(IY,JCLUS)=YPLT(IY,JCLUS)+ARRAY(NJ,IY)
 1428 CONTINUE
      YPLT(IY,JCLUS)=YPLT(IY,JCLUS)/FLOAT(ND)
 1425 CONTINUE
 1420 CONTINUE
      NST=NCLUS2
      READ(20,1413) NCLUS2
      IF(NCLUS2.EQ.'SAME') GO TO 1523
      REREAD 1412,NCLUS2
      DO 1522 JCLUS=1,NCLUS2
      READ(20,1412) NDJ2(JCLUS)
      ND=NDJ2(JCLUS)
      READ(20,1412) (NDJA2(II,JCLUS),II=1,ND)
 1522 CONTINUE
      GO TO 1524
 1523 NCLUS2=NST
 1524 DO 1520 JCLUS=1,NCLUS2
      DO 1525 IY=1,96
      YPLT2(IY,JCLUS)=0.
      ND=NDJ2(JCLUS)
      DO 1528 IY2=1,ND
      NJ=NDJA2(IY2,JCLUS)
      YPLT2(IY,JCLUS)=YPLT2(IY,JCLUS)+ARRAY2(NJ,IY)
 1528 CONTINUE
      YPLT2(IY,JCLUS)=YPLT2(IY,JCLUS)/FLOAT(NDJ)
 1525 CONTINUE
 1520 CONTINUE
C
C              CLUSTER CORRELATIONS
C
      NUMCOR=NCOR(JA)
      WRITE(3,102)
      DO 100 JCOR=1,NUMCOR
      I1=NLIST(JA,JCOR,1)
      I4=I1-NCLUS
      I3=NLIST(JA,JCOR,2)
      I2=I3-NCLUS
  102 FORMAT(/10X,'CLUSTER 1',10X,'CLUSTER 2',20X,'R'/)
      DO 200 I=1,96
      IF(I1.LE.NCLUS) GO TO 4936
      X(I)=YPLT2(I,I4)
      GO TO 4937
 4936 X(I)=YPLT(I,I1)
 4937 CONTINUE
      Y(I)=YPLT2(I,I2)
  200 CONTINUE
      CALL CCOEFF(96,X,Y,R)
      WRITE(3,201) I1,I3,R
  201 FORMAT(12X,I3,16X,I3,18X,G13.4)
  100 CONTINUE
C
C                DAILY CORRELATIONS
C
      WRITE(3,204)
  204 FORMAT(///13X,'DAY',14X,'CLUSTER',21X,'R'/)
      NUMCOR=DCOR(JA)
      DO 300 JCOR=1,NUMCOR
      I1=DLIST(JA,JCOR,1)
      I3=DLIST(JA,JCOR,2)
      I2=I3-NCLUS
      DO 400 I=1,96
      X(I)=ARRAY(I1,I)
      Y(I)=YPLT2(I,I2)
  400 CONTINUE
      CALL CCOEFF(96,X,Y,R)
      WRITE(3,201) I1,I3,R
  300 CONTINUE
      RETURN
      END
      SUBROUTINE CCOEFF(N,X,Y,R)
      DIMENSION X(N),Y(N),TEMP(200)
      XN=FLOAT(N)
      CALL SUM(X,N,A)
      CALL SUM(Y,N,B)
      DO 20 J=1,N
   20 TEMP(J)=X(J)*Y(J)
      CALL SUM(TEMP,N,C)
      DO 21 J=1,N
   21 TEMP(J)=X(J)**2
      CALL SUM(TEMP,N,D)
      DO 22 J=1,N
   22 TEMP(J)=Y(J)**2
      CALL SUM(TEMP,N,E)
      R1=C-A*B/XN
      R2=SQRT((D-(A**2)/XN)*(E-(B**2)/XN))
      R=R1/R2
      RETURN
      END
      SUBROUTINE SUM(X,N,S)
      DIMENSION X(N)
      S=0.
      DO 100 J=1,N
  100 S=S+X(J)
      RETURN
      END
```

```
************************************************************************************
*                              CLUSTERING PROGRAMS
************************************************************************************
       BLOCK DATA
       COMMON NCOR,DCOR,NLIST,DLIST
       INTEGER NLIST(5,9,2),DLIST(5,4,2),NCOR(5),DCOR(5)
       DATA NCOR/6,6,9,9,9/
       DATA DCOR/3,3,3,4,4/
       DATA (NLIST(1,J,1),J=1,6)/1,2,3,4,5,6/,(NLIST(1,J,2),J=1,6)/
      1 4,5,6,5,6,4/,
      A (NLIST(2,J,1),J=1,6)/1,2,3,4,5,6/,(NLIST(2,J,2),J=1,6)/
      2 4,5,6,5,6,4/
       DATA (NLIST(3,J,1),J=1,9)/1,3,1,2,2,2,4,5,6/
       DATA (NLIST(3,J,2),J=1,9)/6,5,4,5,6,4,5,6,4/
       DATA (NLIST(4,J,2),J=1,9)/4,5,6,4,5,6,5,6,4/
       DATA (NLIST(4,J,1),J=1,9)/1,1,2,3,3,3,4,5,6/
       DATA (NLIST(5,J,2),J=1,9)/5,6,5,6,5,7,6,7,5/
       DATA (NLIST(5,J,1),J=1,9)/1,1,2,2,3,4,5,5,7/
       DATA (DLIST(1,J,1),J=1,3)/88,26,49/,(DLIST(1,J,2),J=1,3)/6,5,4/
       DATA (DLIST(2,J,1),J=1,3)/88,26,49/,(DLIST(2,J,2),J=1,3)/6,5,4/
       DATA (DLIST(3,J,1),J=1,3)/26,6,78/,(DLIST(3,J,2),J=1,3)/5,4,4/
       DATA (DLIST(4,J,1),J=1,4)/6,46,45,45/
       DATA (DLIST(4,J,2),J=1,4)/4,6,4,5/
       DATA (DLIST(5,J,1),J=1,4)/88,49,6,37/
       DATA (DLIST(5,J,2),J=1,4)/7,5,5,6/
       END
```

```
*****************************************************************
*                         CLUSTERING PROGRAMS
*****************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:CORAL.DAT [4255,516] 18-May-7
*****************************************************************
ORD
3
64
17 10 30 23 16 9 29 22 15 28 21 14 8 7 55 27 90 83 76 69
62 87 80 73 66 86 79 72 65 3 48 20 85 78 71 64 84 77 70 63
59 58 57 56 52 2 41 13 45 38 51 44 37 50 43 36 49 42 35 31
24 1 34 6
13
89 82 75 68 61 54 47 40 33 26 19 12 5
13
88 81 74 67 60 53 46 39 32 25 18 11 4
3
20
28 27 11 17 26 25 21 14 7 20 4 10 13 6 19 12 5 18 24 3
4
23 16 9 2
4
22 15 8 1
BOS
SAME
SAME
CLE
3
72
90 83 76 69 62 87 85 80 78 73 71 66 64 84 77 70 63 86 79 72
65 88 81 74 67 60 59 57 58 56 52 50 45 43 38 36 51 44 37 49
42 35 31 24 17 10 30 23 16 9 29 28 22 21 15 14 8 7 55 48
41 34 27 20 13 53 46 39 32 25 18 11
5
6 3 2 1 5
13
89 82 75 68 61 54 47 40 33 26 19 12 5
SAME
SEA
3
50
90 83 76 69 62 87 80 86 79 73 66 72 65 85 78 71 64 84 77 70
63 89 82 75 68 61 59 58 57 56 55 52 45 38 51 44 37 50 43 36
49 42 35 48 41 34 54 47 40 33
9
88 81 74 67 60 53 39 32 46
31
31 24 17 10 30 23 16 9 29 22 15 8 28 21 14 7 27 20 13 6
3 2 1 26 19 12 5 25 18 11 4
SAME
```

```
********************************************************************************
*                        CLUSTERING PROGRAMS
********************************************************************************
MIA
4
22
89 82 75 68 61 90 83 76 69 62 87 80 73 66 86 79 72 65 85 78
71 64
43
59 52 45 38 31 24 17 10 3 58 51 4 37 30 23 16 9 2 57 50
43 36 29 22 15 8 1 55 48 41 34 27 20 13 6 54 47 40 33 26
19 12 5
12
84 77 70 63 56 49 42 35 28 21 14 7
13
88 81 74 67 60 53 46 39 32 25 18 11 4
SAME
```

```
*************************************************************************
*                          CLUSTERING PROGRAMS
*************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:AVGPRF.F4 [4255,516] 18-May-7
*************************************************************************
C
C     RUN THIS PROGRAM FOR EACH OF THE MINUTE-BY-MINUTE PROFILE
C        TAPES GENERATED BY OAGPRF
C
      DIMENSION ND(3),WM(1440,0/1),ADR(24),
     1 CONCEN(24),PFS(120),WMS(1440,0/1,3),
     2 TOTAL(120),ADRS(24,3),CONS(24,3)
      DIMENSION PMIXP(24,8),PMIXS(24,8,3)
      DIMENSION NDC(120),MONTHS(8),NNM(8)
      DATA MONTHS/29,31,31,30,28,31,31,30/
      DATA ICL,IAND/"41630,"777777700000/
      DATA NNM/"30142,"30144,"30146,"30150,"30152,"30154,
     1 "30156,"30160/
      INTEGER A1
      WRITE(5,104)
      READ(5,101) IDEV
      WRITE(5,5001)
      READ(5,101) NA1,NA2
C
C            SPECIFY MONTH WHICH IS ON TAPE
C
      WRITE(5,114)
  114 FORMAT('   WHICH TAPE? - '/
     1 '     ENTER AN INTEGER 1.LE.I.LE.8'/
     2 '     1=FEB72 ETC'/)
  101 FORMAT(10I)
      READ(5,101) ITAPE
      WRITE(5,216)
  216 FORMAT('  IN THIS SERIES OF RUNS,'/
     1 '   WHAT IS THE LOWEST TAPE NUMBER YOU ARE USING?'/)
      READ(5,101) NT1
      NADD=0
      IF(ITAPE.EQ.NT1) GO TO 214
      DO 215 JADD=NT1,ITAPE-1
      NADD=NADD+MONTHS(JADD)
  215 CONTINUE
C
C            (NADD+1) = FIRST DAY ON TAPE
C
  214 CONTINUE
      DO 5000 IA=NA1,NA2
 5009 CONTINUE
C
C            READ AIRPORT CODE AND NUMBER OF DAYS FROM TAPE
C
```

```
      READ(IDEV,ERR=5500,END=5007) A1,NDAYS
      WRITE(5,88) A1,NDAYS
   88 FORMAT('  AIRPORT ',A3,' NDAYS 'I4)
C
C            .GET NAME FOR OUTPUT FILE
C
       NAME=(A1.AND.IAND).OR.NNM(ITAPE)
C
C             OPEN OUTPUT FILE
C
      CALL OFILE(21,NAME)
      GO TO 5008
 5007 GO TO 5009
 5008 CONTINUE
  104 FORMAT('  INPUT TAPE DEVICE NUMBER?'/)
  115 FORMAT(A5)
 5001 FORMAT(' LOCATIONS IN LIST OF FIRST AND LAST AIRPORTS?'/)
C
C             GET NAME OF FILE WITH CLUSTER ASSIGNMENTS
C
      NAMEC=(A1.AND.IAND).OR.ICL
C
C             OPEN CLUSTER FILE
C
      CALL IFILE(22,NAMEC)
C
C             READ NUMBER OF CLUSTERS, AND ARRAY
C                OF CLUSTER ASSIGNMENTS
C
      READ(22) NC,NDC
C
C             INITIALIZE ARRAYS
C
      DO 703 IC=1,NC
  703 ND(IC)=0
      DO 1010 II=1,3
      DO 1011 JJ=1,1440
      DO 1011 KK=0,1
      WMS(JJ,KK,II)=0.
 1011 CONTINUE
      DO 1012 JJ=1,24
      ADRS(JJ,II)=0.
      CONS(JJ,II)=0.
      DO 1014 KK=1,8
 1014  PMIXS(JJ,KK,II)=0.
 1012 CONTINUE
 1010 CONTINUE
```

```
      IDAY=NADD
      DO 500 JN=1,NDAYS
C
C             READ ONE DAY'S RECORD FROM TAPE
C
      READ(IDEV) ID,DTOT,WM,ADR,CONCEN,PMIXP,PF
      IDAY=IDAY+1
C
C             ADD DAY TO COUNTER FOR ITS CLUSTER
C
      J=NDC(IDAY)
      ND(J)=ND(J)+1
C
C             ADD MINUTE-BY-MINUTE PROFILE TO SUM
C
      DO 501 NAD=0,1
      DO 501 JJ=1,1440
  501 WMS(JJ,NAD,J)=WMS(JJ,NAD,J)+WM(JJ,NAD)
C
C             ADD HOURLY A/D RATIO AND CONCENTRATION TO SUMS
C
      DO 502 JJ=1,24
      ADRS(JJ,J)=ADRS(JJ,J)+ADR(JJ)
      CONS(JJ,J)=CONS(JJ,J)+CONCEN(JJ)
C
C             ADD POLLUTION MIX TO SUM
C
      DO 503 JJJ=1,8
  503 PMIXS(JJ,JJJ,J)=PMIXS(JJ,JJJ,J)+PMIXP(JJ,JJJ)
  502 CONTINUE
C
C             STORE PEAKING FACTOR IN ARRAY
C
      PFS(IDAY)=PF
C
C             STORE DAILY TOTAL IN ARRAY
C
      TOTAL(IDAY)=DTOT
  500 CONTINUE
C
C             WRITE TO OUTPUT FILE AIRPORT NAME,
C             NUMBER OF DAYS, AND NUMBER OF CLUSTERS
C
      WRITE(21) A1,NDAYS,NC
C
C             WRITE SUMS TO OUTPUT FILE
C
```

```
      CALL WRT(NC,ND,WMS,ADRS,CONS,PMIXS,PFS,TOTAL)
      CALL RELEAS(21)
      CALL RELEAS(22)
 5000 CONTINUE
 5500 CONTINUE
      END
```

```
**********************************************************************
*                        CLUSTERING PROGRAMS
**********************************************************************
      SUBROUTINE WRT(NC,ND,WMS,ADRS,CONS,PMIXS,PFS,TOTAL)
      DIMENSION ND(NC),WMS(1440,0/1,NC),ADRS(24,NC),CONS(24,NC),
     1 PFS(120),TOTAL(120)
      DIMENSION PMIXS(24,8,NC)
      WRITE(21) ND,TOTAL,WMS,ADRS,CONS,PMIXS,PFS
      RETURN
      END
```

```
************************************************************************************
*                          CLUSTERING PROGRAMS
************************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:AVGMRG.F4 [4255,516] 18-May-7(
************************************************************************************
C
C     AVGMRG.F4 MERGES EIGHT DISK FILES WITH SUMMED PROFILES
C
      DIMENSION PMIXS1(24,8,3),PMIXS2(24,8,3)
      DIMENSION N1(3),N2(3),WMS1(1440,0/1,3),WMS2(1440,0/1,3),
     1 ADRS1(24,3),ADRS2(24,3),CONS1(24,3),CONS2(24,3),
     1 PF1(120),PF2(120),
     3 TOTAL1(120),TOTAL2(120)
      DIMENSION AN(31),NNM(8)
      INTEGER AIR,AN,A1,A2
      DATA AN/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
      DATA NNM/"30142,"30144,"30146,"30150,"30152,"30154,
     1 "30156,"30160/
      DATA ICP,IAND/"41640,"777777700000/
      DATA NDAYS/120/
      WRITE(5,101)
  101 FORMAT('  STARTING AND ENDING LOCATIONS IN AIRPORT LIST?'/)
      READ(5,102) NA1,NA2
  102 FORMAT(2I)
      WRITE(5,104)
      READ(5,102) NF
      WRITE(5,103)
  103 FORMAT('  NUMBER OF TAG FOR FIRST INPUT FILE?'/)
      READ(5,102) NF1
      NFA=NF1-1
  104 FORMAT('  HOW MANY INPUT FILES?'/)
      DO 2000 IA=NA1,NA2
      AIR=AN(IA).AND.IAND
      NAMEO=AIR.OR.ICP
C
C             OPEN OUTPUT FILE
C
      CALL OFILE(20,NAMEO)
      NAMEI=AIR.OR.NNM(NF1)
C
C             OPEN FIRST INPUT FILE
C
      CALL IFILE(21,NAMEI)
C
C             READ FIRST INPUT FILE
C
      READ(21) A1,ND1,NC1
      CALL SREAD(NC1,N1,WMS1,ADRS1,CONS1,
     2 PMIXS1,PF1,TOTAL1)
C
C             RELEASE FIRST INPUT FILE
C
      CALL RELEAS(21)
C
C             LOOP OVER REMAINING INPUT FILES
C
      DO 4340 JF=2,NF
      JFI=JF+NFA
      NAMEI=AIR.OR.NNM(JFI)
C
C             OPEN
C
      CALL IFILE(21,NAMEI)
C
C             READ
C
      READ(21) A2,ND2,NC2
      IF(NC1.NE.NC2) GO TO 1003
      IF(A1.NE.A2) GO TO 1000
      ND1=ND1+ND2
      CALL SREAD(NC2,N2,WMS2,ADRS2,CONS2,
     2 PMIXS2,PF2,TOTAL2)
C
C             RELFASE
C
      CALL RELEAS(21)
C
C             SUM FOR CLUSTERS
C
      DO 50 I=1,NC1
      N1(I)=N1(I)+N2(I)
      DO 51 I2=1,1440
      DO 51 I3=0,1
   51 WMS1(I2,I3,I)=WMS1(I2,I3,I)+WMS2(I2,I3,I)
      DO 52 I2=1,24
      ADRS1(I2,I)=ADRS1(I2,I)+ADRS2(I2,I)
      CONS1(I2,I)=CONS1(I2,I)+CONS2(I2,I)
      DO 53 I3=1,8
      PMIXS1(I2,I3,I)=PMIXS1(I2,I3,I)+PMIXS2(I2,I3,I)
   53 CONTINUE
   52 CONTINUE
   50 CONTINUE
C
C     INSERT DAILY TOTALS AND PEAKING FACTORS
```

```
C      FROM INPUT FILE INTO COMPREHENSIVE LIST
C
       DO 60 JP=1,NDAYS
       IF(TOTAL2(JP).GT.TOTAL1(JP)) TOTAL1(JP)=TOTAL2(JP)
       IF(PF2(JP).GT.PF1(JP)) PF1(JP)=PF2(JP)
   60  CONTINUE
C       WRITE(5,34)
   34  FORMAT(' WRITE')
 4340  CONTINUE
C
C               AVERAGE
C
       NTOT=N1(1)+N1(2)+N1(3)
       IF(NTOT.NE.NDAYS) WRITE(5,253)
  253  FORMAT(2X,'   NDAYS = 'I5)
       DO 4253 IC=1,3
       XD=FLOAT(N1(IC))
       DO 12 IM=1,1440
        DO 12 IAD=0,1
       WMS1(IM,IAD,IC)=WMS1(IM,IAD,IC)/XD
   12  CONTINUE
       DO 13 IH=1,24
       ADRS1(IH,IC)=ADRS1(IH,IC)/XD
       CONS1(IH,IC)=CONS1(IH,IC)/XD
       DO 15 IPX=1,8
   15  PMIXS1(IH,IPX,IC)=PMIXS1(IH,IPX,IC)/XD
   13  CONTINUE
 4253  CONTINUE
C
C               WRITE OUTPUT FILE
C
       WRITE(20) A1,ND1,NC1
       CALL WRT(NC1,N1,WMS1,ADRS1,CONS1,
      1 PMISX1,PF1,TOTAL1)
C
C               RELEASE OUTPUT FILE
C
       CALL RELEAS(20)
 2000  CONTINUE
       GO TO 1001
 1003  WRITE(5,1004)
 1004  FORMAT(' VALUES FOR NUMBER OF CLUSTERS DO NOT AGREE'/)
       GO TO 1001
 1000  WRITE(5,1002)
 1002  FORMAT(' AIRPORT CODES DO NOT MATCH'/)
 1001  CONTINUE
       END
```

```
************************************************************************
*                          CLUSTERING PROGRAMS
************************************************************************
      SUBROUTINE WRT(NC,ND,WMS,ADRS,CONS,PMIXS,PFS,TOTAL)
      DIMENSION ND(NC),WMS(1440,0/1,NC),ADRS(24,NC),CONS(24,NC),
     1 PFS(120),TOTAL(120)
      DIMENSION PMIXS(24,8,NC)
      WRITE(20) ND,TOTAL,WMS,ADRS,CONS,PMIXS,PFS
      RETURN
      END
      SUBROUTINE SREAD(NC,ND,WMS,ADRS,CONS,PMIXS,PFS,TOTAL)
      DIMENSION ND(NC),WMS(1440,0/1,NC),ADRS(24,NC),CONS(24,NC),
     1 PFS(120),TOTAL(120)
        DIMENSION PMIXS(24,8,NC)
      READ(21) ND,TOTAL,WMS,ADRS,CONS,PMIXS,PFS
      RETURN
      END
```

APPENDIX C

```
*************************************************************************
*                        WEATHER PROGRAMS
*************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:EXTWEA.F4 [4255,516] 18-May-7
*************************************************************************
C                   EXTWEA.F4
C
C            THIS PROGRAM EXTRACTS WEATHER DATA (CEILING AND
C            VISIBILITY) FOR TWO YEARS (72 AND 73) FROM A TDF-14 TAPE
C
      DIMENSION IA(7),AR(20)
      NW=0
      NR=0
C
C            GET INPUT AND OUTPUT DEVICE NUMBERS
C
      WRITE(5,300)
  300 FORMAT('  INPUT TAPE NUMBER?'/)
      READ(5,301) IDEV
  301 FORMAT(I)
      WRITE(5,302)
  302 FORMAT('  OUTPUT TAPE NUMBER?'/)
      READ(5,301) IOUT
C
C            READ A RECORD
C
  200 READ(IDEV,10,ERR=2000,END=1000) IA
   10 FORMAT(I5,4I2,1X,2I3)
      NR=NR+1
      IF((NR/50000*50000).EQ.NR) WRITE(5,307) NR
  307 FORMAT('    NR='I10)
C
C            IS THIS A RECORD WE WANT (72 OR 73 DATA)
C
      IF(IA(2).EQ.72.OR.IA(2).EQ.73) GO TO 100
      GO TO 200
C
C         WRITE A RECORD TO OUTPUT TAPE
C
  100 WRITE(IOUT,10) IA
      NW=NW+1
      IF((NW/20000*20000).EQ.NW) WRITE(5,308) NW
  308 FORMAT('  NW='I10)
      GO TO 200
 1000 END FILE IOUT
C
C            WRITE TO TERMINAL NUMBER OF WEATHER RECORDS EXTRACTED
C
      WRITE(5,102) NW
  102 FORMAT('  WEATHER RECORDS EXTRACTED ='I10/)
```

```
      GO TO 400
 2000 CONTINUE
C
C           IF READ ERROR, REREAD IN (20A1) FORMAT
C
      REREAD 110,AR
  110 FORMAT(20A1)
      NR=NR+1
C
C            WRITE ERROR BEARING RECORD TO TERMINAL
C
      WRITE(5,210) AR,NR
  210 FORMAT(5X,20A1,'   INPUT RECORD  'I10/)
C
C           IS THE ERROR-BEARING RECORD ONE WE WANT?
C
      IF(AR(7).EQ.'2'.OR.AR(7).EQ.'3') GO TO 500
      GO TO 200
  500 NW=NW+1
C
C           WRITE AN ERROR-BEARING RECORD TO OUTPUT TAPE
C
      WRITE(IOUT,110) AR
      WRITE(5,310) NW
  310 FORMAT(20X,'  OUTPUT RECORD'I10)
      GO TO 200
  400 CONTINUE
      END
```

```
***********************************************************************
*                            WEATHER PROGRAMS
***********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:WEACOM.F4 [4255,516] 18-May-7
***********************************************************************
C                       WEACOM.F4
C
C             THIS PROGRAM COMPACTS WEATHER DATA EXTRACTED BY
C             WEACOM.F4.  IN CASE OF READ ERRORS, CEILING AND
C             VISIBILITY ARE SET TO 0.
C             AN END-OF-FILE IS WRITTEN AFTER DATA
C             FOR EACH STATION.
C
      DIMENSION IA(4),ICLG(8),IVIS(8)
C
C    THE NEXT TWO STATEMENTS ARE SPECIAL PLEADING
C    FOR TWO MISSING RECORDS
C
      LOGICAL ILOG
      DATA ILOG/.FALSE./
      LIM=8
      IFILE=1
      IW=0
      IR=0
      IW=0
C
C             GET INPUT AND OUTPUT DEVICE NUMBERS
C
      WRITE(5,30)
   30 FORMAT('  INPUT TAPE NUMBER?'/)
      READ(5,31) IDEV
   31 FORMAT(I)
      WRITE(5,32)
   32 FORMAT('  OUTPUT TAPE NUMBER?'/)
      READ(5,31) IOUT
C
C             READ A RECORD
C
  200 READ (IDEV,10,ERR=140,END=100) IA,IHR,ICLG(1),IVIS(1)
   10 FORMAT(I5,4I2,1X,2I3)
      GO TO 144
  140 REREAD 410,IA
  410 FORMAT(I5,3I2)
      ICLG(1)=0
      IVIS(1)=0
  144 CONTINUE
      IR=IR+1
      IF(IR.EQ.1) GO TO 801
C
C             IF DATA READ IS FOR A DIFFERENT STATION THAN
```

```
C              THE LAST RECORD, WRITE AN END-OF-FILE ON THE
C              OUTPUT TAPE.
C
         IF(IA(1).EQ.NSTA) GO TO 801
         END FILE IOUT
         WRITE(5,47) NSTA,IFILE
         IFILE=IFILE+1
    47   FORMAT('  STATION ',I6,' IS ON FILE 'I4)
   801   CONTINUE
         NSTA=IA(1)
   800   CONTINUE
C
C              READ SEVEN MORE RECORDS - COMPLETES INPUT
C              OF DATA FOR ONE DAY (8 OBSERVATIONS)
C
C
C       THE NEXT SEVEN STATEMENTS ARE SPECIAL PLEADING
C       FOR TWO MISSING RECORDS
C
         IF(IA(1).NE.24233) GO TO 888
         IF(IA(2).NE.72) GO TO 888
         IF(IA(3).NE.12) GO TO 888
         IF(IA(4).NE.31) GO TO 888
         ILOG=.TRUE.
   888   CONTINUE
         IF(ILOG) LIM=6
          DO 20 J=2,LIM
          READ(IDEV,11,ERR=240) ICLG(J),IVIS(J)
    11   FORMAT(14X,2I3)
         GO TO 244
   240   REREAD 410,IA
         ICLG(J)=0
         IVIS(J)=0
   244   CONTINUE
    20   CONTINUE
C
C
C       THE NEXT TWO RECORDS ARE SPECIAL PLEADING
C       FOR TWO MISSING RECORDS
C
         IF(ILOG) LIM=8
         IF(ILOG) ILOG=.FALSE.
C              WRITE OUT ONE RECORD FOR DAY
C
         WRITE(IOUT) IA,ICLG,IVIS
         IW=IW+1
         GO TO 200
```

```
  100 CONTINUE
      END FILE IOUT
      WRITE(5,47) NSTA,IFILE
      WRITE(5,48) IR,IW
   48 FORMAT(/' RECORDS READ'I10/' RECORDS WRITTEN'I10)
      END
```

```
***********************************************************************
*                          WEATHER PROGRAMS
***********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:VFRIFR.F4 [4255,516] 18-May-7
***********************************************************************
C                  VFRIFR.F4
C           THIS PROGRAM SEPARATES DAYS (FOR TWO
C           YEARS) INTO VFR AND IFR DAYS.  INPUT TAPE WRITTEN
C           BY WEACOM.F4
C
      DIMENSION IAIR(30)
      INTEGER AIR
      DIMENSION IA(4),ICL(8),IVS(8),LIST(731)
      DATA IWE,IAND/"53612,"777777700000/
        DATA IAIR/'DAL','MIA','TPA','MSY','IAH','PHL','DCA','ATL','CL
     1 'JAX',
     1 'STL','LGA','EWR','BOS','CLE','MKE','MSP','HNL','DEN','LAS',
     1 'LAX','SFO','SLC',
     1 'SEA','BAL','IAD','JFK','PIT','ORD','DTW'/
      DATA NA/30/
      DATA NDAYS/731/
C
C           GET INPUT DEVICE NUMBER
C
      WRITE(5,30)
   30 FORMAT(' INPUT DEVICE NUMBER?'/)
      READ(5,11) IDEV
   11 FORMAT(I)
C
C      READ A RECORD
C
      IAPT=0
 1120      IDAY=0
      IAPT=IAPT+1
      DO 1130 I=1,NDAYS
 1130 LIST(I)=0
      IF(IAPT.GT.NA) GO TO 2000
 1100 READ(IDEV,ERR=1000,END=1000) IA,ICL,IVS
      IF(IDAY.GT.0) GO TO 112
      AIR=IAIR(IAPT)
      NAME=(AIR.AND.IAND) .OR. IWE
C
      WRITE(5,140)  IA(1),AIR
  140 FORMAT('  STATION NUMBER ='I8,10X,A3/)
      IOUT=20
      CALL OFILE(IOUT,NAME)
  112 IDAY=IDAY+1
      IC=0
      IV=0
C
```

```
**********************************************************************
*                             WEATHER PROGRAMS
**********************************************************************
C             CHECK CEILING AND VISIBILITY
C
      DO 300 JC=3,8
      IF(ICL(JC).GE.15) IC=IC+1
      IF(IVS(JC).GE.30) IV=IV+1
  300 CONTINUE
C
C             DECIDE IF DAY IS VFR OR IFR
C
      IF(MINO(IC,IV).GE.5) LIST(IDAY)=1
C
C     IF(LIST(DAY)=1,  DAY IS VFR.
C         OTHERWISE, LIST(DAY)=0 AND DAY IS IFR.
C
      GO TO 1100
 1000 CONTINUE
C
C             WRITE NUMBER OF DAYS FOUND TO TERMINAL FOR A CHECK -
C             SHOULD BE EQUAL TO NDAYS
C
      IF(IDAY.NE.NDAYS)       WRITE(5,180) IDAY
  180 FORMAT('  NUMBER OF DAYS FOUND FOR THIS AIRPORT:'/
     1 20X,I6)
C
C       WRITE LIST OF VFR AND IFR DAYS ON DISK FILE
C
      WRITE(IOUT) AIR,LIST
      END FILE IOUT
      CALL RELEAS(IOUT)
      GO TO 1120
 2000 CONTINUE
      END
```

APPENDIX D

```
*********************************************************************
*                            TOWER PROGRAMS
*********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:TOWRD.F4  [4255,516] 18-May-7
*********************************************************************
C            TOWRD.F4
C
      DIMENSION NPT(31),AN(31),NARRAY(3294),NUM(183)
      DATA AN,NPT/'BOS','DCA','BAL','EWR','JFK',
     1 'LGA','PHL','PIT','IAD','FLL','JAX','MIA',
     2 'TPA','ATL','CLT','ORD','DTW','MSP','CLE','MKE','STL','MSY',
     3 'DAL','IAH','DEN','SLC','LAX','SFO','LAS','SEA','HNL',7,18,20,
     4 32,33,46,48,57,70,75,78,90,95,110,136,151,164,171,184,201,213,
     5 236,242,256,269,287,305,316,331,343/
      INTEGER AN
      WRITE(5,201)
  201 FORMAT('   OUTPUT FILE NAME?'/)
      READ(5,103) ANAME
  103 FORMAT(A5)
      IOUT=20
      CALL OFILE(IOUT,ANAME)
  102 FORMAT(30I)
      WRITE(5,203)
  203 FORMAT('  INPUT TAPE NUMBER?'/)
      READ(5,1102) ITAPE
      WRITE(5,101)
  101 FORMAT('  RANGE OF AIRPORTS(BY NUMBER)  ON THIS TAPE?'
     1 ' - EG, 1,100'/)
      READ(5,1102) NAT1,NAT2
 1102 FORMAT(2I)
      NR=0
  400 NR=NR+1
      IF(NPT(NR).LT.NAT1) GO TO 400
      DO 1000 IR=NAT1,NAT2
      IF(IR.EQ.NPT(NR)) GO TO 500
      READ(ITAPE,501) NUM
  501 FORMAT(I7)
      WRITE(5,10) IR
   10 FORMAT(I8)
      GO TO 1000
  500 READ(ITAPE,502) NARRAY
  502 FORMAT(18I7)
      WRITE(IOUT) AN(NR),NARRAY
      WRITE(5,11) IR
   11 FORMAT(10X,I8)
      NR=NR+1
 1000 CONTINUE
  600 CONTINUE
      END FILE IOUT
       END
```

```
**********************************************************************
*                             TOWER PROGRAMS
**********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:TOWRD3.F4 [4255,516] 18-May-7
**********************************************************************
C               TOWRD3.F4
C
      DIMENSION NPT(31),AN(31),NARRAY(3294),NUM(183)
      DATA AN,NPT/'BOS','DCA','BAL','EWR','JFK',
     1 'LGA','PHL','PIT','IAD','FLL','JAX','MIA',
     2 'TPA','ATL','CLT','ORD','DTW','MSP','CLE','MKE','STL','MSY',
     3 'DAL','IAH','DEN','SLC','LAX','SFO','LAS','SEA','HNL',7,18,20,
     4 32,33,46,48,58,71,76,79,91,96,113,139,154,167,174,187,204,216,
     5 239,245,259,272,291,309,320,337,349/
      DIMENSION AAA(126)
      INTEGER AN
      WRITE(5,201)
  201 FORMAT('   OUTPUT FILE NAME?'/)
      READ(5,103) ANAME
  103 FORMAT(A5)
      IOUT=20
      CALL OFILE(IOUT,ANAME)
  102 FORMAT(30I)
      WRITE(5,203)
  203 FORMAT('  INPUT TAPE NUMBER?'/)
      READ(5,1102) ITAPE
      WRITE(5,101)
  101 FORMAT('  RANGE OF AIRPORTS(BY NUMBER)  ON THIS TAPE?'
     1 ' - EG, 1,100'/)
      READ(5,1102) NAT1,NAT2
 1102 FORMAT(2I)
      NR=0
  400 NR=NR+1
      IF(NPT(NR).LT.NAT1) GO TO 400
      DO 1000 IR=NAT1,NAT2
      IF(IR.EQ.NPT(NR)) GO TO 500
      READ(ITAPE,501) NUM
  501 FORMAT(I7)
      WRITE(5,10) IR
   10 FORMAT(I8)
      GO TO 1000
  500 READ(ITAPE,502,ERR=573) NARRAY
  502 FORMAT(18I7)
      WRITE(IOUT) AN(NR),NARRAY
      GO TO 576
  573 REREAD 574,AAA
  574 FORMAT(126A1)
      WRITE(5,575) AAA
  575 FORMAT(5X,126A1)
  576 CONTINUE
```

```
      WRITE(5,11) IR
   11 FORMAT(10X,I8)
      NR=NR+1
 1000 CONTINUE
  600 CONTINUE
      END FILE IOUT
       END
```

```
*********************************************************************
*                            TOWER PROGRAMS
*********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:TOWMRG.F4 [4255,516] 18-May-7
*********************************************************************
C           TOWMRG.F4
C
      DIMENSION NARRAY(3294),IARRAY(3294)
      DATA IAND,ITW/"777777700000,"52256/
      INTEGER A1,A2
      WRITE(5,100)
  100 FORMAT(' NAME TWO TOWER INPUT FILES'/)
      READ(5,101) NAME1,NAME2
  101 FORMAT(A5,1X,A5)
      WRITE(5,102)
  102 FORMAT('  HOW MANY AIRPORTS ON THESE FILES?'/)
      READ(5,103) NA
  103 FORMAT(I)
      CALL IFILE(20,NAME1)
      CALL IFILE(21,NAME2)
      DO 55 JA=1,NA
      READ(20) A1,IARRAY
      READ(21) A2,NARRAY
      WRITE(5,104) A1,A2
  104 FORMAT(/10X,A3,5X,A3/)
      NAME3=(A1.AND.IAND).OR.ITW
      CALL OFILE(22,NAME3)
      WRITE(22) A1,IARRAY,NARRAY
      ENDFILE 22
      CALL RELEAS(22)
   55 CONTINUE
      END
```

```
*************************************************************************
*                             TOWER PROGRAMS
*************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:TOWER.F4    [4255,516] 18-May-7
*************************************************************************
C                                 TOWER
C
C               THIS PROGRAM SEPARATES TOWER TRAFFIC FOR
C               VFR AND IFR GROUPS OF DAYS;  CALCULATES
C               NON-SCHEDULED MIXES FOR EACH DAY;  COMPUTES
C               VFR AND IFR DISTRIBUTIONS;  AND CALCULATES
C               MEANS FOR AC, AT, GA, AND MIL TRAFFIC.
C
      DIMENSION MNAC(0/1),MNAT(0/1),MNGA(0/1),MNMIL(0/1),
     1 MNUM(0/1)
      DIMENSION LIST(731),VS(731),VN(731),
     1 NARRAY(6588),XMIX(0/1,8,4),MIX(8,731)
      DATA ITW,IWE,ITO,IAND/"52256,"53612,"52236,"777777700000/
      INTEGER AN(31)
      DATA AN/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
      DATA IIV/"44654/
      DATA (XMIX(0,K,1),K=1,8)/0.,0.,0.,.05,0.,.15,.80,0.0/
      DATA (XMIX(0,K,2),K=1,8)/0.,0.,0.,.05,0.,.15,.80,0.0/
      DATA (XMIX(1,K,1),K=1,8)/0.,0.,0.,.01,0.,.03,.96,0.0/
      DATA (XMIX(1,K,2),K=1,8)/0.,0.,0.,.01,0.0,.03,.96,0.0/
      DATA (XMIX(0,K,3),K=1,8)/0.,0.,0.,.20,0.0,.10,.30,.40/
      DATA (XMIX(0,K,4),K=1,8)/0.,0.,0.,.20,0.0,.10,.30,.40/
      DATA (XMIX(1,K,3),K=1,8)/0.,0.,0.,.20,0.0,.10,.30,.40/
      DATA (XMIX(1,K,4),K=1,8)/0.,0.,0.,.20,0.0,.10,.30,.40/
      COMMON/DEVS/IOUT
      REAL MMIX(8)
      REAL MIX,MNAC,MNAT,MNGA,MNMIL
      DATA NDAYS,NAPT/731,31/
      CALL ERRSET(50)
C
C               HOW MANY AIRPORTS THIS RUN?
C
                WRITE(5,30)
      READ(5,21) NA
      IA=0
  500 IA=IA+1
      IF(NA.EQ.NAPT) GO TO 131
C
C               GET NAME OF AIRPORT
C
      WRITE(5,330)
  330 FORMAT(///'    WHAT AIRPORT?'/)
```

```
      READ(5,335) IAIR
  335 FORMAT(A3)
      GO TO 231
  131 IAIR=AN(IA)
  231 CONTINUE
C
C             GET NAME OF FILE WITH LIST OF VFR AND
C             IFR DAYS (WEATHER) FOR THIS AIRPORT
C             AND FILE WITH TOWER DATA;
C             ALSO NAME OUTPUT FILE
C
      INAME=IAIR.AND.IAND
      NAMET=INAME.OR.ITW
      NAMEW=INAME.OR.IWE
      NAME2=INAME.OR.ITO
      NAMEI=INAME.OR.IIV
C
C             OPEN TOWER DATA FILE
C
      CALL IFILE(22,NAMET)
      IDEV=22
   21 FORMAT(101)
      IOUT=21
C
C             OPEN OUTPUT FILES
C
       CALL OFILE(IOUT,NAME2)
   30 FORMAT(' HOW MANY AIRPORTS?'/)
      IDSK=20
      IOUT2=23
      CALL OFILE(IOUT2,NAMEI)
C
C             OPEN WEATHER FILE
C
      CALL IFILE(IDSK,NAMEW)
      READ(IDSK) IAIR1,LIST
      READ(IDEV) IAIR2,NARRAY
C
C             CHECK THAT FILES ARE INDEED FOR THE AIRPORT WANTED
C
      IF(IAIR1.NE.IAIR) GO TO 600
      IF(IAIR2.NE.IAIR) GO TO 600
      WRITE(IOUT) IAIR
      WRITE(5,723) IAIR
  723 FORMAT(5X,'AIRPORT: ',A3)
      DO 743 III=0,1
      MNAC(III)=0.
      MNAT(III)=0.
      MNGA(III)=0.
      MNMIL(III)=0.
      MNUM(III)=0
  743 CONTINUE
      DO 749 III=1,8
  749 MMIX(III)=0.
          DO 700 JD=1,NDAYS
      N1=(JD-1)*9+1
      N2=N1+1
      N3=N1+2
      N4=N1+3
      N5=N1+4
      N6=N1+5
      N7=N1+6
      N8=N1+7
      VS(JD)=FLOAT(NARRAY(N1)+NARRAY(N2))
      VN(JD)=NARRAY(N3)+NARRAY(N4)+NARRAY(N6)+NARRAY(N7)
      L1=LIST(JD)
C
C             CALCULATE NON-SCHEDULED MIX
C
      DO 800 IMIX=1,8
      MIX(IMIX,JD)=0.
      IF(VN(JD).LT.1.0E-04) GO TO 881
      MIX(IMIX,JD)=(XMIX(L1,IMIX,1)*NARRAY(N3)+
     1  XMIX(L1,IMIX,2)*NARRAY(N6)+XMIX(L1,IMIX,3)*NARRAY(N4)+
     2 XMIX(L1,IMIX,4)*NARRAY(N7))/VN(JD)
  881 MMIX(IMIX)=MMIX(IMIX)+MIX(IMIX,JD)
  800 CONTINUE
      MNUM(L1)=MNUM(L1)+1
      MNAC(L1)=MNAC(L1)+NARRAY(N1)
      MNAT(L1)=MNAT(L1)+NARRAY(N2)
      MNGA(L1)=MNGA(L1)+NARRAY(N3)+NARRAY(N6)
      MNMIL(L1)=MNMIL(L1)+NARRAY(N4)+NARRAY(N7)
  700 CONTINUE
C
C             CALCULATE 2-YEAR MEAN FOR NON-SCHED MIX
C
      DO 704 III=1,8
  704 MMIX(III)=MMIX(III)/NDAYS
C
C             CALCULATE MEANS FOR AC, AT, GA, AND MIL
C
      DO 745 LLL=0,1
      IF(MNUM(LLL).EQ.0) GO TO 745
      MNAC(LLL)=MNAC(LLL)/MNUM(LLL)
      MNAT(LLL)=MNAT(LLL)/MNUM(LLL)
      MNGA(LLL)=MNGA(LLL)/MNUM(LLL)
      MNMIL(LLL)=MNMIL(LLL)/MNUM(LLL)
  745 CONTINUE
      WRITE(IOUT) MNUM,MNAT,MNAC,MNGA,MNMIL,MMIX
C
C             CALL SUBROUTINE TO CALCULATE DISTRIBUTIONS, ETC
C
      CALL XFRDIS(NDAYS,VS,VN,MIX,LIST,MNUM)
      CALL RELEAS(IDSK)
      CALL RELEAS(IOUT)
      CALL RELEAS(IOUT2)
      CALL RELEAS(IDEV)
      IF(IA.LT.NA) GO TO 500
      GO TO 603
  600 WRITE(5,601) IA,AN1,AN2
  601 FORMAT(' FOR IA ='I4,' AN1 = ',A3,'AND AN2 = ',A3/)
  603 CONTINUE
      END
      SUBROUTINE XFRDIS(NDAYS,VS,VN,MIX,LIST,MNUM)
      DIMENSION LIST(NDAYS),VS(NDAYS),
     1 VN(NDAYS),MNUM(0/1),IVINT(5),JVINT(5),
     2 MIX(8,NDAYS),IORD1(731),IORD2(731)
      DIMENSION IVN(0/1,5)
      REAL MIX,MIXA,IVINT,JVINT
      REAL IVN
      COMMON/ORDERS/ILIST
      DIMENSION NPI2(0/1)
      DIMENSION NPI(0/1),NSIG(0/1),PI(0/1,5),SIG(0/1,5),NV2(5),
     1 NV1(5),IBOX(5,5),NST(5,5,250),VSA(5,5),VNA(5,5),
     2 MIXA(8,5,5),AMIX(8)
      DATA NPI,NSIG,(SIG(0,I),I=1,3)/4*3,.5,.85,1.0/
      DATA (SIG(1,I),I=1,3)/.5,.85,1.0/
      COMMON/DEVS/IOUT
C
C            THIS SUBROUTINE SEPARATES TOWER TRAFFIC
C            INTO VFR AND IFR DAYS AND FOR EACH OF THESE
C            SETS OF DAYS:
C               (1) CALCULATES NON-SCHEDULED MIXES
C               (2) CALCULATES DISTRIBUTIONS
C               (3) AVERAGES SCHEDULED VOLUMES AND
C                   NON-SCHEDULED VOLUMES AND MIXES
C                   FOR EACH OF THE BOXES (I,J)
C                   DEFINED BY THE DISTRIBUTION LEVELS
C               (4) CALCULATES INTERVALS IV(I) AND JV(J)
C                   S.T. IV(I) IS THE LARGEST SCHEDULED
C                   VOLUME FOR THE I-TH SET OF BOXES, AND
C                   JV(J) IS THE LARGEST NON-SCHEDULED VOLUME
C                   FOR THE J-TH SET OF BOXES
C
      DATA (PI(0,I),I=1,3)/.5,.85,1.0/
      DATA (PI(1,I),I=1,3)/.5,.85,1.0/
C
C            ILIST=0  :  IFR CALCULATIONS
C            ILIST=1  :  VFR CALCULATIONS
C
      DO 1000 ILIST=0,1
      NPI2(ILIST)=NPI(ILIST)
C
C            ICNT = NUMBER OF DAYS WITH THE TYPE OF WEATHER
C            CURRENTLY SELECTED
C
      ICNT=MNUM(ILIST)
      XN=FLOAT(ICNT)
C
C            IF THERE ARE FEWER THAN 10 OF THESE DAYS,
C            GO TO  188  AND DO AN ALTERNATIVE CALCULATION
C
      NIB=NPI(ILIST)
      NJB=NSIG(ILIST)
      DO 210 I=1,NIB
      DO 210 J=1,NJB
  210 IBOX(I,J)=0
      IF(ICNT.EQ.0) GO TO 588
      IF(ICNT.LE.10) GO TO 188
C
C            ORDER THE DAYS BY SCHEDULED VOLUMES
C
      CALL ORDER(NDAYS,VS,LIST,ICNT,IORD1)
C
C            ORDER THE DAYS BY NON-SCHEDULED VOLUMES
C
      CALL ORDER(NDAYS,VN,LIST,ICNT,IORD2)
C
C            GET DISTRIBUTION LEVELS FOR SCHEDULED VOLUMES
C
      CALL DISLEV(NPI(ILIST),PI,XN,NV1)
C
C            GET DISTRIBUTION LEVELS FOR NON-SCHEDULED VOLUMES
C
      CALL DISLEV(NSIG(ILIST),SIG,XN,NV2)
C
C            FILL BOXES WITH ZEROES
C
C
C            COUNT THE DAYS THAT GO INTO EACH BOX
C
      DO 201 IALF=1,ICNT
      DO 202 IBET=1,ICNT
      IF(IORD1(IALF).NE.IOPD2(IBET)) GO TO 205
      I=0
  206      I=I+1
      IF(IALF.GT.NV1(I)) GO TO 206
      J=0
  207 J=J+1
      IF(IBET.GT.NV2(J)) GO TO 207
      IBOX(I,J)=IBOX(I,J)+1
      IORDR=IBOX(I,J)
      IF(IORDR.GT.250) WRITE(5,9112) IORDR
 9112 FORMAT('NST NEEDS RE-DIMENSIONING  -'/
     1 '  IBOX.GT.250  = 'I5/)
      NST(I,J,IORDR)=IORD2(IBET)
  205 CONTINUE
  202 CONTINUE
  201 CONTINUE
C
C   CALCULATE AVERAGES FOR EACH I,J FOR:
C              (1) SCHEDULED VOLUMES [AVS]
C              (2) NON-SCHEDULED VOLUMES [AVN]
C              (3) NON-SCHEDULED MIXES [AMIX]
C
      DO 400 I=1,NIB
      DO 400 J=1,NJB
      AVS=0.
      AVN=0.
      DO 412 JJ=1,5
  412 AMIX(JJ)=0.
      ND=IBOX(I,J)
      AMAXI=0.
      AMAXJ=0.
      IF(ND.EQ.0) GO TO 424
      DO 401 LD=1,ND
      NDAY=NST(I,J,LD)
      AVN=AVN+VN(NDAY)
      AVS=AVS+VS(NDAY)
      DO 413 JJ=1,8
  413 AMIX(JJ)=AMIX(JJ)+VN(NDAY)*MIX(JJ,NDAY)
  401 CONTINUE
      AVS=AVS/ND
      AVN=AVN/ND
      IF(AVN.LT.1.0E-04) GO TO 424
      DO 414 JJ=1,8
  414 AMIX(JJ)=AMIX(JJ)/(ND*AVN)
  424 CONTINUE
      VSA(I,J)=AVS
      VNA(I,J)=AVN
      DO 415 JJ=1,8
  415       MIXA(JJ,I,J)=AMIX(JJ)
  400 CONTINUE
C
C              GET INTERVALS FOR SCHEDULED VOLUMES
C
      DO 420 I=1,NIB
      I2=NV1(I)
      NDAY=IORD1(I2)
      IVINT(I)=VS(NDAY)
  420 CONTINUE
C
C              GET INTERVALS FOR NON-SCHEDULED VOLUMES
C
```

```
      DO 430 J=1,NJB
      J2=NV2(J)
      NDAY=IORD2(J2)
      JVINT(J)=VN(NDAY)
  430 CONTINUE
      GO TO 288
C
C             IF FEWER THAN 10 DAYS, DO THESE CALCULATIONS
C             INSTEAD OF THOSE ABOVE
C
  188 CONTINUE
      NPI2(ILIST)=1
      NIB=1
      NJB=1
      I=1
      J=1
      DO 417 JJ=1,8
  417 AMIX(JJ)=0.
      AVS=0.
      AVN=0.
      AMAXI=0.
      AMAXJ=0.
      DO 501 JD=1,NDAYS
      IF(LIST(JD).NE.ILIST) GO TO 502
      AVS=AVS+VS(JD)
      AVN=AVN+VN(JD)
      AMAXI=AMAX1(AMAXI,VS(JD))
      AMAXJ=AMAX1(AMAXJ,VN(JD))
      DO 418 JJ=1,8
  418 AMIX(JJ)=AMIX(JJ)+VN(JD)*MIX(JJ,JD)
  502 CONTINUE
  501 CONTINUE
      VSA(1,1)=AVS/XN
      VNA(1,1)=AVN/XN
      DO 419 JJ=1,8
      IF(AVN.LT.1.0E-04) GO TO 419
      AMIX(JJ)=AMIX(JJ)/AVN
  419 MIXA(JJ,1,1)=AMIX(JJ)
      IVINT(1)=AMAXI
      JVINT(1)=AMAXJ
      IBOX(I,J)=MNUM(ILIST)
      GO TO 288
  588 NIB=0
      NJB=0
      WRITE(5,473) ILIST,ICNT
  473 FORMAT(/5X'WEATHER='I2,'   NDAYS='I2/)
  288 CONTINUE
```

```
      DO 479  I=1,NIB
  479 IVN(ILIST,I)=IVINT(I)
      WRITE(IOUT) ILIST,NIB,NJB,VSA,VNA,MIXA,IBOX,IVINT,JVINT
 1000 CONTINUE
      WRITE(23) MNUM,NPI2,IVN
      RETURN
      END
      SUBROUTINE DISLEV(N,AR,X,NAR)
      DIMENSION AR(0/1,N),NAR(N)
      COMMON/ORDERS/IL
C
C            CALCULATE DISTRIBUTION LEVELS
C
      DO 40 J=1,N
      XV=AR(IL,J)*X
      NAR(J)=IFIX(XV)
      XC=FLOAT(NAR(J))
      IF((XV-XC).GT.1.0E-03) NAR(J)=NAR(J)+1
   40 CONTINUE
      RETURN
      END
```

```
*********************************************************************
*                           TOWER PROGRAMS
*********************************************************************
      SUBROUTINE ORDER(ND,V,LIST,IC,IORD)
      DIMENSION V(ND),LIST(ND),IORD(IC)
      COMMON/ORDERS/IL
C
C            PICK OUT THE ELEMENTS <V(IV)> OF [V]
C            FOR WHICH LIST(IV)=IL  - INDICATES VFR OR IFR DAYS
C            AND STORE THEIR INDICES [IV] IN ARRAY IORD
C
C
      IV=0
      I=0
  200 I=I+1
      IF(I.GT.IC) GO TO 300
  100 IV=IV+1
      IF(IV.GT.ND) GO TO 300
      IF(LIST(IV).NE.IL) GO TO 100
      IORD(I)=IV
      GO TO 200
  300 CONTINUE
C
C                   REORDER THE INDICES BY INCREASING VALUES OF V(I)
C
      I=0
      ISAVE=0
  800      I=ISAVE+1
      ISAVE=I
      I=I-1
  600 I=I+1
      IF(I.LT.1) GO TO 800
      IF((1+1).GT.IC) GO TO 500
      I1=IORD(I)
      I2=IORD(I+1)
      IF(V(I1).LE.V(I2)) GO TO 800
      IORD(I+1)=I1
      IORD(I)=I2
      I=I-2
      GO TO 600
  500 CONTINUE
      RETURN
      END
```

APPENDIX E

```
***********************************************************************
*                     ANNUALIZATION PROGRAMS
***********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:CLSFRQ.F4 [4255,516] 18-MAY-76
***********************************************************************
C             CLSFRQ.F4
C
C     THIS PROGRAM COMPUTES CLUSTER FREQUENCIES AT EACH DISTRIBUTICN
C     LEVEL, AS DEFINED BY PI(I).
C     DISTRIBUTION LEVELS FROM TOWER DATA (IJV)
C     ARE NOT CURRENTLY IN USE,
C
      DIMENSION IORD(12Ø),PI(3),NV(3)
      DATA NPI/3/,PI/.5,.85,1.Ø/
       DIMENSION NDC(12Ø),SVOL(12Ø),VOL(12Ø),IJV(Ø/1,5),NS(Ø/1,5,5),
     1 MNUM(Ø/1),
     2 NIV(Ø/1),NSUM(Ø/1,5),FV(Ø/1,5,5),NII(Ø/1)
      REAL MNAT(Ø/1),MNAC(Ø/1)
      DATA ITO/"52236/
      DATA ICL,IIV/"4163Ø,"44654/
      REAL IJV
      INTEGER AP(31),AIR
       DATA AP/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SEA','HNL'/
      DATA ICF,IAND/"41614,"7777777ØØØØØ/
      DATA NDAYS/12Ø/
      WRITE(5,293)
  293 FORMAT('   LOCATIONS IN LIST OF FIRST AND LAST AIRPORTS?'/)
      READ(5,294) NA1,NA2
  294 FORMAT(2I)
C
C             LOOP OVER AIRPORTS
C
      DO 1111 NA=NA1,NA2
      AIR=AP(NA).AND.IAND
      NAME=AIR.OR.ICF
      NAME2=AIR.OR.IIV
      NAME3=AIR.OR.ICL
C
C             OPEN OUTPUT FILE
C
      CALL OFILE(22,NAME)
C
C             OPEN INPUT FILES
C
      CALL IFILE(21,NAME2)
      CALL IFILE(2Ø,NAME3)
C
```

```
C               READ CLUSTER ASSIGNMENTS
C
      READ(2Ø) NC,NDC,VOL
C
C               READ TOWER DISTRIBUTION DATA
C
      READ(21) NIV,NII,IJV
      DO 839 JN=1,5
      DO 839 JM=1,5
      DO 839 JL=Ø,1
  839 NS(JL,JM,JN)=Ø
C
C               ORDER DAYS BY OAG VOLUME
C
      CALL ORDER(NDAYS,VOL,IORD)
      XN=FLOAT(NDAYS)
C
C               GET DISTIBUTION LEVELS (OAG VOLUMES)
C
      CALL DISLEV(NPI,PI,XN,NV)
C
C               LOOP OVER WEATHER
C
      DO 2ØØ ILIST=Ø,1
      NI=NII(ILIST)
      IF(NIV(ILIST).LT.1Ø) NI=1
C
C               NDC(NDAYS) ASSOCIATES A CLUSTER NUMBER
C               WITH EACH  DAY
C
C
C               COUNT THE DAYS IN EACH CLUSTER,
C               AT EACH VOLUME LEVEL
C
        DO 1ØØ IDAY=1,NDAYS
      IC=NDC(IDAY)
      II=Ø
   88 II=II+1
      IF(II.EQ.NI) GO TO 188
      IF(IORD(IDAY).GT.NV(II)) GO TO 88
  188 NS(ILIST,II,IC)=NS(ILIST,II,IC)+1
  1ØØ CONTINUE
C
C               SUM THE DAYS FOR EACH LEVEL
C               FOR ALL CLUSTERS
C
      DO 4ØØ J=1,NI
      NSUM(ILIST,J)=Ø
      DO 3ØØ I=1,NC
      NSUM(ILIST,J)=NSUM(ILIST,J)+NS(ILIST,J,I)
  3ØØ CONTINUE
C
C             CALCULATE CLUSTER FREQUENCIES
C
      DO 4ØØ I=1,NC
      A1=NS(ILIST,J,I)
      A2=NSUM(ILIST,J)
      F1=A1/A2
      FV(ILIST,J,I)=F1
  4ØØ CONTINUE
      WRITE(5,912) AIR
  912 FORMAT(/1ØX,'AIRPORT:'A3)
      WRITE(5,911) ((I,J,FV(ILIST,J,I),I=1,NC),J=1,NI)
  911 FORMAT(2X,2I6,G12.4)
      WRITE(22) NC,NI
C
C             WRITE
C
  2ØØ CONTINUE
C
C             WRITE
C
      WRITE(22) FV
      DO 111 IDEV=2Ø,22,2
      CALL RELEAS(IDEV)
  111 CONTINUE
 1111 CONTINUE
      END
      SUBROUTINE ORDER(ND,V,IORD)
      DIMENSION V(ND),IORD(ND)
      DO 3ØØ IV=1,ND
  3ØØ IORD(IV)=IV
C
C                               REORDER THE INDICES BY INCREASING VALUES OF V(I)
C
      I=Ø
      ISAVE=Ø
  8ØØ I=ISAVE+1
      ISAVE=I
      I=I-1
  6ØØ I=I+1
      IF(I.LT.1) GO TO 8ØØ
      IF((I+1).GT.ND) GO TO 5ØØ
      I1=IORD(I)
      I2=IORD(I+1)
      IF(V(I1).LE.V(I2)) GO TO 8ØØ
      IORD(I+1)=I1
      IORD(I)=I2
      I=I-2
      GO TO 6ØØ
  5ØØ CONTINUE
      RETURN
      END
      SUBROUTINE DISLEV(N,AR,X,NAR)
      DIMENSION AR(N),NAR(N)
C
C             CALCULATE DISTRIBUTION LEVELS
C
      DO 40 J=1,N
      XV=AR(J)*X
      NAR(J)=IFIX(XV)
      XC=FLOAT(NAR(J))
      IF((XV-XC).GT.1.0E-03) NAR(J)=NAR(J)+1
   40 CONTINUE
      RETURN
      END
```

```
***********************************************************************
*                       ANNUALIZATION PROGRAMS
***********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:APMFG.F4   [4255,516] 18-MAY-76
***********************************************************************
C              APMFG.F4
C
C     THIS PROGRAM GENERATES FILES FOR INPUT INTO
C     THE APM.
C     FOUR(4) KINDS OF FILES (SEE BELOW).
C       1. DAILY (AVERAGED FOR YEAR)
C       2. HEADER
C       3. ANNUALIZATION
C       4. PROFILE (CLUSTERED)
C
      DIMENSION MNUM(Ø/1),NR(3),VSA(5,5),VNA(5,5),IBOX(5,5)
      DIMENSION ND(3),TOT(12Ø),VOL(144Ø,2,3),ADR(24,3),
     1 CON(24,3),PMIX(24,8,3),PFS(12Ø),VH(6Ø,2),PM(8)
      DIMENSION IAN(31)
      DATA IAND,IDF,IHF,IAF,IPF/"7777777ØØØØØ,"42214,"44214,
     1 "4Ø614,"5Ø214/
      DATA ICP/"4164Ø/
      DATA ITO/"52236/
      DATA IAN/'BOS','DCA','BAL','EWR','JFK','LGA','IAH','PHL',
     1 'PIT','IAD','FLL','JAX','MIA','MKE','TPA','ATL','CLT',
     2 'ORD','DTW','MSP','CLE','STL','MSY','DAL','DEN','SLC',
     3 'LAX','SFO','LAS','SFA','HNL'/
      DATA NDAYS/12Ø/
      INTEGER WTHR(24)
      REAL MIXA(8,5,5)
      REAL MNAT(Ø/1),MNAC(Ø/1),MNGA(Ø/1),MNMIL(Ø/1),MMIX(8)
      INTEGER DF,HF,AF,PF,TO,CP,CF
      DATA DF,HF,AF,PF/16,17,18,19/
      DATA TO/2Ø/,CP/21/
      DATA CF/22/,ICF/"41614/
      DIMENSION FV(Ø/1,5,5)
      DO 2Ø4 IDEV=16,19
      CALL ASSDEV(IDEV,'DSK')
  2Ø4 CONTINUE
      WRITE(5,631)
  631 FORMAT('  LIST LOCATIONS OF FIRST AND LAST AIRPORTS?'/)
      READ(5,632) NA1,NA2
  632 FORMAT(2I)
      DO 1ØØØ IA=NA1,NA2
      INM=IAN(IA).AND.IAND
      NAME1=INM.OR.IDF
      NAME2=INM.OR.IHF
      NAME3=INM.OR.IAF
      NAME4=INM.OR.IPF
      CALL OFILE(DF,NAME1)
      CALL OFILE(HF,NAME2)
      CALL OFILE(AF,NAME3)
      CALL OFILE(PF,NAME4)
      NAME=INM,OR.ITO
      CALL IFILE(TO,NAME)
      NAME=INM,OR.ICP
      CALL IFILE(CP,NAME)
      NAME=INM,OR.ICF
      CALL IFILE(CF,NAME)
      READ(CF) NC1,NI1
      READ(CF) NC,NI
      IF(NC1.NE.NC) WRITE(5,804) NC1,NC
  804 FORMAT(/5X,' NUMBER OF CLUSTERS NOT CONSISTENT ON CF FILE'/
     1 10X,2I5)
      IF(NI1.NE.NI) WRITE(5,805) NI1,NI
  805 FORMAT(/5X,' # OF SCHED INTERVALS NOT CONSISTENT ON CF FILE'/
     1 10X,2I5)
      READ(CF) FV
      CALL RELEAS(CF)
C
C             READ CLUSTERED PROFILES
C
      READ(CP) IA1,NDTOT,NC
      WRITE(5,105) IA1,NDTOT,NC
  105 FORMAT(5X,A3,5X,'DAYS ='I5,5X,'NC ='I5/)
      READ(CP) ND,TOT,VOL,ADR,CON,PMIX,PFS
C
C             WRITE PROFILE FILE
C
      DTOT=FLOAT(NDTOT)
      DO 100 IPR=1,3
      WRITE(PF) IPR
      DO 101 IH=1,24
      DO 241 IP=1,8
  241 PM(IP)=PMIX(IH,IP,IPR)
      AD=ADR(IH,IPR)
      CONC=CON(IH,IPR)
      WRITE(PF) IH,PM,AD,CONC
      IS=(IH-1)*60+1
      IFIN=IS+59
      IMIN=0
      ARR=0.
      DEP=0.
      DO 144 INOW=IS,IFIN
      IMIN=IMIN+1
      DO 143 IAD=1,2
  143 VH(IMIN,IAD)=VOL(INOW,IAD,IPR)
```

```
      ARR=ARR+VH(IMIN,1)
      DEP=DEP+VH(IMIN,2)
  144 CONTINUE
      WRITE(PF) ARR,DEP
      WRITE(PF) VH
  101 CONTINUE
  100 CONTINUE
      CALL RELEAS(PF)
C
C             GET THE PEAKING FACTOR FOR THIS AIRPORT
C
      PFM=0.
      DO 221 I12=1,NDAYS
       PFM=AMAX1(PFM,PFS(I12))
  221 CONTINUE
C
C             GET 2-YEAR AVERAGES FOR SCHED,NØN-SCHED
C             VOLUMES
C
      READ(TO) IA2
      READ(TO) MNUM,MNAT,MNAC,MNGA,MNMIL,MMIX
      MN=MNUM(Ø)+MNUM(1)
      XMN=FLOAT(MN)
      VSY=MNUM(Ø)*(MNAT(Ø)+MNAC(Ø))+MNUM(1)*
     1 (MNAC(1)+MNAT(1))
      VSY=VSY/XMN
      VNY=MNUM(Ø)*(MNGA(Ø)+MNMIL(Ø))+MNUM(1)*
     1 (MNGA(1)+MNMIL(1))
      VNY=VNY/XMN
C
C             AVERAGE CLUSTERED PROFILFS AND WRITE DAILY FILE
C
      DO 222 II=1,24
  222 WTHR(II)=1
      WRITE(DF) VSY,VNY,MMIX,WTHR
      DO 3Ø1 IH=1,24
      DO 341 IP=1,8
      PM(IP)=Ø.
      DO 342 I2=1,3
  342 PM(IP)=PM(IP)+PMIX(IH,IP,I2)*ND(I2)
      PM(IP)=PM(IP)/DTOT
  341 CONTINUE
      AD=Ø.
      DO 343 I2=1,3
      AD=ADR(IH,I2)*ND(I2)
      CONC=CON(IH,I2)*ND(I2)
  343 CONTINUE
```

```
      AD=AD/DTOT
      CONC=CONC/DTOT
      WRITE(DF) IH,PM,AD,CONC
      IS=(IH-1)*6Ø+1
      IFIN=IS+59
      IMIN=Ø
      ARR=Ø.
      DEP=Ø.
      DO 344 INOW=IS,IFIN
      IMIN=IMIN+1
      DO 345 IAD=1,2
      TEMP=Ø.
      DO 346 INT=1,3
  346 TEMP=TEMP+VOL(INOW,IAD,INT)*ND(INT)
      VH(IMIN,IAD)=TEMP/DTOT
  345 CONTINUE
      ARR=ARR+VH(IMIN,1)
       DEP=DEP+VH(IMIN,2)
  344 CONTINUE
      WRITE(DF) ARR,DEP
      WRITE(DF) VH
  301 CONTINUE
      CALL RELEAS(DF)
      NBOX=Ø
      NRUNS=Ø
C
C             WRITE ANNUALIZATION FILE
C
      DO 15ØØ I276=1,2
      READ(TO) ILIST,NIB,NJB,VSA,VNA,MIXA,IBOX
      NBOXS=Ø
      IF(NIB.EQ.Ø) GO TO 157Ø
      IF(NJB.EQ.Ø) GO TO 157Ø
      DO 7 IB=1,NIB
      DO 7 JB=1,NJB
    7 NBOXS=NBOXS+IBOX(IB,JB)
      NBOX=NBOX+NBOXS
      XBOXS=NBOXS
      DO 1 IR=1,NIB
      DO 1 JR=1,NJB
      IF(VSA(IR,JR).GT.1.ØE-Ø4) GO TO 2
      IF(VNA(IR,JR).LT.1.ØE-Ø4) GO TO 1
    2 NRUNS=NRUNS+NC
      DO 5 NCC=1,NC
    5 NR(NCC)=NRUNS-(NC-NCC)
      WRITE(AF) NC,NR
      WRITE(AF) VSA(IR,JR),VNA(IR,JR),(MIXA(IM,IR,JR),IM=1,8),ILIST
      DO 4 IC=1,NC
      W=FLOAT(IBOX(IR,JR))*FV(ILIST,IR,IC)/XBOXS
      W=W*(XBOXS/731.)*365.
      WRITE(AF) NR(IC),IC,W
    4 CONTINUE
    1 CONTINUE
 15ØØ CONTINUE
C
C              WRITE HEADER FILE
C
      WRITE(HF) IAN(IA),PFM,MNUM(1),MNUM(Ø),NRUNS
      WRITE(HF) MNAC(1),MNAT(1),MNGA(1),MNMIL(1),
     1 MNAC(Ø),MNAT(Ø),MNGA(Ø),MNMIL(Ø)
C
C              CHECK/NOCHECK
C
      ICHECK=NBOX/MN
      IF(ICHECK,EQ,1) GO TO 6
      WRITE(5,6Ø1) IAN(IA)
  6Ø1 FORMAT(5X,'  NOCHECK',5X,A3)
      GO TO 8
    6 WRITE(5,6Ø2) IAN(IA)
  6Ø2 FORMAT(5X,'  CHECK',5X,A3)
    8 CONTINUE
 1ØØØ CONTINUE
      END
```

```
*****************************************************************
*                      ANNUALIZATION PROGRAMS
*****************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:FPRINT.F4 [4255,516] 18-MAY-76
*****************************************************************
C             FPRINT.F4
C
C     THIS PROGRAM OFFERS THE OPPORTUNITY OF OBTAINING
C     PRINTED OUTPUT OF THE CONTENTS OF THE UNFORMATTED
C     FILES WRITTEN BY APMFG.
C
      DATA IAND,IDF,IHF,IAF,IPF/"7777777ØØØØØ,"42214,"44214,
     1 "4Ø614,"5Ø214/
      COMMON IAND,IDF,IHF,IAF,IPF,IAIR
      WRITE(5,1ØØ)
  1ØØ FORMAT(' AIRPORT?'/)
      READ(5,2ØØ) IAIR
  2ØØ FORMAT(A3)
    1 WRITE(5,3ØØ)
  3ØØ FORMAT('  WHAT FILE WOULD YOU LIKE PRINTED?  --'/
     1 '    DAILY,HEADER,ANNUALIZATION,PROFILE,NONE?'/)
      READ(5,4ØØ) A
  4ØØ FORMAT(A1)
      IF(A.EQ.'D') GO TO 1Ø
      IF(A.EQ.'H') GO TO 2Ø
      IF(A.EQ.'A') GO TO 3Ø
      IF(A.EQ.'P') GO TO 4Ø
      GO TO 5Ø
   1Ø CALL DPR
      GO TO 1
   2Ø CALL HPR
      GO TO 1
   3Ø CALL APR
      GO TO 1
   4Ø CALL PPR
      GO TO 1
   5Ø CONTINUE
      END
      SUBROUTINE DPR
      DIMENSION XM(8),SM(8),VH(6Ø,2)
      INTEGER WTH(24)
      COMMON IAND,IDF,I1,I2,I3,IAIR
      CALL ASSDEV(16,'DSK')
      IAN=(IAIR.AND.IAND).OR.IDF
      CALL IFILE(16,IAN)
      READ(16) VS,VN,XM,WTH
      WRITE(3,1ØØ) VS,VN,XM,WTH
      DO 1 JH=1,24
      READ(16) IH,SM,ADR,CONCEN
      WRITE(3,2ØØ) IH,SM,ADR,CONCEN
      READ(16) ARR,DEP
      WRITE(3,3ØØ) ARR,DEP
      READ(16) VH
      WRITE(3,4ØØ) ((VH(J,1),VH(J,2)),J=1,6Ø)
    1 CONTINUE
  1ØØ FORMAT(/1ØX,'VS= ',F12.Ø,1ØX,'VN= ',F12.Ø/8G12.4/5X,24I2)
  2ØØ FORMAT(5X,'HOUR ='I3/
     1 8G12.4/
     2 5X,'ADR =',F12.6,5X,'CONCEN =',F12.6)
  3ØØ FORMAT('  ARRIVALS =',F12.6,5X,' DEPARTURES ='F12.6)
  4ØØ FORMAT(5(5X,2F1Ø.4))
      RETURN
      END
```

```
*********************************************************
*                        ANNUALIZATION PROGRAMS
*********************************************************
      SUBROUTINE HPR
      DIMENSION VM(8),N(4)
      COMMON IAND.I1,IHF,I2,I3,IAIR
      CALL ASSDEV(16,'DSK')
      IAN=(IAIR,AND.IAND).OR.IHF
      CALL IFILE(16,IAN)
      READ(16) IAIR,PF,NDV,NDI,NRUNS
      WRITE(3,1ØØ) IAIR,PF,NDV,NDI,NRUNS
      READ(16) VM
      WRITE(3,2ØØ) VM
  1ØØ FORMAT(///1ØX,'AIRPORT:'A3/
     1 1ØX,'PEAKING FACTOR:'G14.4/
     2 1ØX,'VFR DAYS:'I5/
     3 1ØX,'IFR DAYS:'I5/
     4 1ØX,'ANNUALIZATION RUNS:'I5///)
  2ØØ FORMAT(2ØX'MEANS'/5X8G13.3)
  3ØØ FORMAT(///1ØX,I5,1ØX,3I5)
  4ØØ FORMAT(/1ØX'SCHEDULED VOLUME:'G13.2/
     1 1ØX,'NON-SHEDULED VOLUME:'G13.2/
     2 1ØX,'MIX:'8G13.2/
     3 1ØX,'WEATHER:',I3)
  5ØØ FORMAT(1ØX,'N:',I6,5X,'I:',I6,5X,'WEIGHT:'G13.2)
      CALL RELEAS(16)
      RETURN
      END
```

```
*************************************************************
*                      ANNUALIZATION PROGRAMS
*************************************************************
      SUBROUTINE APR
      DIMENSION VM(8),N(4)
      COMMON IAND,I1,IHF,IAF,I2,IAIR
      CALL ASSDEV(16,'DSK')
      IAN=(IAIR,AND.IAND).OR.IHF
      CALL IFILE(16,IAN)
      READ(16) IAIR,PF,NDV,NDI,NRUNS
      READ(16) VM
  3ØØ FORMAT(///1ØX,I5,1ØX,3I5)
  4ØØ FORMAT(/1ØX'SCHEDULED VOLUME:'G13.2/
     1 1ØX,'NON-SHEDULED VOLUME:'G13.2/
     2 1ØX,'MIX:'8G13.2/
     3 1ØX,'WEATHER:',I3)
  5ØØ FORMAT(1ØX,'N:',I6,5X,'I:',I6,5X,'WEIGHT:'G13.2)
      CALL RELEAS(16)
      IAN=(IAIR,AND.IAND).OR.IAF
      CALL IFILE(16,IAN)
      NQ=NRUNS/3
      IF((NQ-1)/4*4.EQ.(NQ-1)) WRITE(3,6ØØ)
  6ØØ FORMAT(1H1)
      DO 25 JH=1,NQ
      READ(16) N
      WRITE(3,3ØØ) N
      READ(16) V1,V2,VM,IW
      WRITE(3,4ØØ) V1,V2,VM,IW
      DO 26 J=1,3
      READ(16) IN,IP,W
      WRITE(3,5ØØ) IN,IP,W
   26 CONTINUE
   25 CONTINUE
      RETURN
      END
```

```
************************************************************************
*                        ANNUALIZATION PROGRAMS
************************************************************************
      SUBROUTINE PPR
      DIMENSION XM(8),SM(8),VH(60,2)
      COMMON IAND,I1,I2,I3,IPF,IAIR
      INTEGER WTH(24)
      CALL ASSDEV(16,'DSK')
      IAN=(IAIR,AND.IAND).OR.IPF
      CALL IFILE(16,IAN)
      NC=3
      DO 5 JC=1,NC
      IF(JC.GT.1) WRITE(3,600)
  600 FORMAT(1H1)
      READ(16) NP
      WRITE(3,100) NP
      DO 1 JH=1,24
      READ(16) IH,SM,ADR,CONCEN
      WRITE(3,200) IH,SM,ADR,CONCEN
      READ(16) ARR,DEP
      WRITE(3,300) ARR,DEP
      READ(16) VH
      WRITE(3,400) ((VH(J,1),VH(J,2)),J=1,60)
    1 CONTINUE
    5 CONTINUE
  100 FORMAT(//10X'PROFILE'I5//)
  200 FORMAT(5X,'HOUR ='I3/
     1 8G12.4/
     2 5X,'ADR =',F12.6,5X,'CONCEN =',F12.6)
  300 FORMAT('  ARRIVALS =',F12.6,5X,' DEPARTURES ='F12.6)
  400 FORMAT(5(5X,2F10.4))
      RETURN
      END
```

APPENDIX F

```
*****************************************************************
*                     INCIDENTAL PROGRAMS
*****************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:WEATHR.F4 [4255,516] 18-MAY-76
*****************************************************************
C                   WEATHR
C               THIS PROGRAM SEPARATES DAYS (FOR TWO
C               YEARS) INTO VFR AND IFR DAYS.  INPUT TAPE WRITTEN
C               BY WEACOM.F4.  ALSO ASSIGNS VFR OR IFR STATUS
C               TO EACH HOUR OF EACH DAY.
C
      DIMENSION IAIR(3Ø)
      INTEGER AIR
      DIMENSION IA(4),ICL(8),IVS(8),IVA(8,731)
      DATA IWP,IAND/"5354Ø,"7777777ØØØØØ/
        DATA IAIR/'DAL','MIA','TPA','MSY','IAH','PHL','DCA','ATL','CLT'
     1 'JAX',
     1 'STL','LGA','EWR','BOS','CLE','MKE','MSP','HNL','DEN','LAS',
     1 'LAX','SFO','SLC',
     1 'SEA','BAL','IAD','JFK','PIT','ORD','DTW'/
C
C               GET INPUT DEVICE NUMBER
C
      WRITE(5,3Ø)
   3Ø FORMAT(' INPUT DEVICE NUMBER?'/)
      READ(5,11) IDEV
   11 FORMAT(I)
C
C        READ A RECORD
C
      IAPT=Ø
 112Ø         IDAY=Ø
      IAPT=IAPT+1
      DO 113Ø I=1,731
      DO 1134 I2=1,8
 1134 IVA(I2,I)=Ø
 113Ø CONTINUE
      IF(IAPT.GT.3Ø) GO TO 2ØØØ
 11ØØ READ(IDEV,ERR=1ØØØ,END=1ØØØ) IA,ICL,IVS
      IF(IDAY.GT.Ø) GO TO 112
      AIR=IAIR(IAPT)
      NAME=(AIR.AND.IAND) .OR. IWP
C
      WRITE(5,14Ø)  IA(1),AIR
  14Ø FORMAT('  STATION NUMBER ='I8,1ØX,A3/)
      IOUT=2Ø
      CALL OFILE(IOUT,NAME)
  112 IDAY=IDAY+1
C
C               ASSIGN VFR OR IFR STATUS TO EACH
```

```
****************************************************************************
*                              INCIDENTAL PROGRAMS
****************************************************************************
C                       OBSERVATION FOR THE DAY
C
      DO 1136 JJ=1,8
      IF((ICL(JJ).GE.15).AND.(IVS(JJ).GE.3Ø)) IVA(JJ,IDAY)=1
 1136 CONTINUE
      GO TO 11ØØ
 1ØØØ CONTINUE
C
C               WRITE NUMBFR OF DAYS FOUND TO TERMINAL FOR A CHECK -
C               SHOULD BF 731
C
      IF(IDAY.NE.731)       WRITE(5,18Ø) IDAY
  18Ø FORMAT('  NUMBER OF DAYS FOUND FOR THIS AIRPORT:'/
     1 2ØX,I6)
C
C         WRITE LIST OF VFR AND IFR DAYS ON DISK FILE
C
      WRITE(IOUT) AIR,IVA
      END FILE IOUT
      CALL RELEAS(IOUT)
      GO TO 112Ø
 2ØØØ CONTINUE
      END
```

```
************************************************************
*                         INCIDENTAL PROGRAMS
************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:WEATH2.F4 [4255,516] 18-MAY-76
************************************************************
C        WEATH2.F4
C
      DIMENSION N(3),L(3),IVA(8,731),IV2(24,122/152)
      DATA N/'EWRWØ','JFKWØ','LGAWØ'/
      DATA L/'EWRWP','JFKWP','LGAWP'/
      DO 2ØØ I=1,3
      NAME=N(I)
      CALL IFILE(2Ø,NAME)
      READ(2Ø) IAIR,IVA
      DO 3ØØ JJ=122,152
      DO 2Ø1 J2=1,8
      J3=(J2-1)*3+1
      DO 7Ø J4=J3,J3+2
      IV2(J4,JJ)=IVA(J2,JJ)
   7Ø CONTINUE
  2Ø1 CONTINUE
  3ØØ CONTINUE
      NAME=L(I)
      CALL OFILE(21,NAME)
      WRITE(21,1Ø1) IAIR
  1Ø1 FORMAT(2X,A3)
      WRITE(21,1Ø2) IV2
  1Ø2 FORMAT(24I2)
      CALL RELEAS(2Ø)
      CALL RELEAS(21)
  2ØØ CONTINUE
      END
```

```
*************************************************************************
*                          INCIDENTAL PROGRAMS
*************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:WEATH4.F4 [4255,516] 18-MAY-76
*************************************************************************
C                    WEATH4.F4
C
C                    THIS PROGRAM PRINTS OUT CEILING AND
C              VISIBILITY DATA FOR EWR,LGA,JFK(OR ANY OTHER
C              AIRPORT FOR THAT MATTER) FROM AN INPUT TAPE.  YOU
C              HAVE TO RUN THEM ONE AT A TIME AS THEY ARE NOT
C              CONTIGUOUS ON THE WEATHER TAPE
C              FILE 12 IS LGA
C              FILE 13 IS EWR
C              FILE 27 IS JFK
C
      INTEGER AIR
      DIMENSION IA(4),ICL(8),IVS(8),IVA(8,731)
C
C              GET INPUT DEVICE NUMBER
C
      WRITE(5,3Ø)
   3Ø FORMAT(' INPUT DEVICE NUMBER?'/)
      READ(5,11) IDEV
   11 FORMAT(I)
      WRITE(5,1Ø9)
  1Ø9 FORMAT('  WHAT AIRPORT?'/)
      READ(5,119) AIR
  119 FORMAT(A3)
      WRITE(3,244) AIR
  244 FORMAT(1H1///6ØX,A3//)
      WRITE(3,245)
  245 FORMAT(5X,'DATE',22X,'CEILING',27X,'VISIBILITY'////)
C
C        READ A RECORD
C
 112Ø         IDAY=Ø
      DO 113Ø I=1,731
      DO 1134 I2=1,8
 1134 IVA(I2,I)=Ø
 113Ø CONTINUE
 11ØØ READ(IDEV,ERR=1ØØØ,END=1ØØØ) IA,ICL,IVS
      IF(IDAY.GT.Ø) GO TO 112
      WRITE(5,14Ø)  IA(1),AIR
  14Ø FORMAT('  STATION NUMBER ='I8,1ØX,A3/)
  112 IDAY=IDAY+1
C
C              PRINT OUT CEILING AND VISIBILITY IF
C              122.LE.IDAY.LE.152
C
```

```
      IDATE=IDAY-121
      IF(IDATE.LT.1) GO TO 112
      IF(IDATE.GT.31) GO TO 1ØØØ
      WRITE(3,243) IDATE,ICL,IVS
  243 FORMAT(2X,'MAY',I3,',1972'5X,8I4,5X,8I4)
C
 1136 CONTINUE
      GO TO 11ØØ
 1ØØØ CONTINUE
C
 2ØØØ CONTINUE
      END
```

```
***************************************************************************
*                             INCIDENTAL PROGRAMS
***************************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:TOWMAY.F4 [4255,516] 18-MAY-76
***************************************************************************
C             TOWMAY.F4
C
      DIMENSION LIST(731),VS(731),VN(731),
     1 NARRAY(6588),XMIX(Ø/1,8,4),MIX(8,731)
      DIMENSION WTH(24),XM(8)
      DIMENSION VOL(Ø/1,2,2)
      DATA ITW,IWE,ITO,IAND/"52256,"53612,"52236."777777ØØØØØØ/
      DATA ID1/"42142/
      DATA (XMIX(Ø,K,1),K=1,8)/Ø.,Ø.,Ø.,.Ø5,Ø.Ø,.15,.8Ø,Ø.Ø/
      DATA (XMIX(Ø,K,2),K=1,8)/Ø.,Ø.,Ø.,.Ø5.Ø.Ø,.15,.8Ø,Ø.Ø/
      DATA (XMIX(1,K,1),K=1,8)/Ø.,Ø.,Ø.,.Ø1,.Ø,.Ø3,.96,Ø.Ø/
      DATA (XMIX(1,K,2),K=1,8)/Ø.,Ø.,Ø.,.Ø1.Ø.Ø,.Ø3,.96,Ø.Ø/
      DATA (XMIX(Ø,K,3),K=1,8)/Ø.,Ø.,Ø.,.2Ø,Ø.Ø,.1Ø,.3Ø,.4Ø/
      DATA (XMIX(Ø,K,4),K=1,8)/Ø.,Ø.,Ø.,.2Ø,Ø.Ø,.1Ø,.3Ø,.4Ø/
      DATA (XMIX(1,K,3),K=1,8)/Ø.,Ø.,Ø.,.2Ø,Ø.Ø,.1Ø,.3Ø,.4Ø/
      DATA (XMIX(1,K,4),K=1,8)/Ø.,Ø.,Ø.,.2Ø,Ø.Ø,.1Ø,.3Ø,.4Ø/
      COMMON/DEVS/IOUT
      REAL MIX
C
C             HOW MANY AIRPORTS THIS RUN?
C
              WRITE(5,3Ø)
      READ(5,21) NA
      IA=Ø
  5ØØ IA=IA+1
C
C             GET NAME OF AIRPORT
C
      WRITE(5,33Ø)
  33Ø FORMAT(///'   WHAT AIRPORT?'/)
      READ(5,335) IAIR
  335 FORMAT(A3)
C
C             GET NAME OF FILE WITH LIST OF VFR AND
C             IFR DAYS (WEATHER) FOR THIS AIRPORT
C             AND FILE WITH TOWER DATA;
C             ALSO NAME OUTPUT FILE
C
      INAME=IAIR.AND.IAND
      NAMET=INAME.OR.ITW
      NAMEW=INAME.OR.IWE
      NAME2=INAME.OR.ID1
C
C             OPEN TOWER DATA FILE
C
```

```
      CALL IFILE(22,NAMET)
      IDEV=22
   21 FORMAT(10I)
      IOUT=21
C
C             OPEN OUTPUT FILES
C
       CALL OFILE(IOUT,NAME2)
   30 FORMAT(' HOW MANY AIRPORTS?'/)
      IDSK=20
C
C             OPEN WEATHER FILE
C
      CALL IFILE(IDSK,NAMEW)
      READ(IDSK) IAIR1,LIST
      READ(IDEV) IAIR2,NARRAY
C
C             CHECK THAT FILES ARE INDEED FOR THE AIRPORT WANTED
C
      IF(IAIR1.NE.IAIR) GO TO 600
      IF(IAIR2.NE.IAIR) GO TO 600
      WRITE(IOUT) IAIR
            DO 700 JD=1,152
      N1=(JD-1)*9+1
      N2=N1+1
      N3=N1+2
      N4=N1+3
      N5=N1+4
      N6=N1+5
      N7=N1+6
      N8=N1+7
      VS(JD)=FLOAT(NARRAY(N1)+NARRAY(N2))
      VN(JD)=NARRAY(N3)+NARRAY(N4)+NARRAY(N6)+NARRAY(N7)
      L1=LIST(JD)
C
C             CALCULATE NON-SCHEDULED MIX
C
      DO 800 IMIX=1,8
      MIX(IMIX,JD)=(XMIX(L1,IMIX,1)*NARRAY(N3)+
     1  XMIX(L1,IMIX,2)*NARRAY(N6)+XMIX(L1,IMIX,3)*NARRAY(N4)+
     2 XMIX(L1,IMIX,4)*NARRAY(N7))/VN(JD)
  800 CONTINUE
  700 CONTINUE
      DO 804 II=122,152
      VS1=VS(II)
      VN1=VN(II)
      DO 805 II2=1,8
```

```
 8Ø5 XM(II2)=MIX(II2,II)
     DO 8Ø6 II2=1,24
 8Ø6 WTH(II2)=LIST(II)
     IDAY=II-121
     WRITE(IOUT) IDAY,VS1,VN1,XM,WTH
 8Ø4 CONTINUE
     CALL RELEAS(IDSK)
     CALL RELEAS(IOUT)
     CALL RELEAS(IDEV)
     IF(IA.LT.NA) GO TO 5ØØ
     GO TO 6Ø3
 6ØØ WRITE(5,6Ø1) IA,AN1,AN2
 6Ø1 FORMAT(' FOR IA ='I4,' AN1 = ',A3,'AND AN2 = ',A3/)
 6Ø3 CONTINUE
     END
```

```
**********************************************************************
*                         INCIDENTAL PROGRAMS
**********************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:DAILY.F4  [4255,516] 18-MAY-76
**********************************************************************
C               DAILY.F4
C
        DIMENSION AN(3),XM(8),WTH(24)
        DIMENSION SM(8),VH(60,2)
        INTEGER AN
        DIMENSION VOL(1440,2),ADR(24),CONCEN(24),
      1 PMIXP(24,8)
        DATA AN/'EWR','JFK','LGA'/
        DATA ID1,ID2,IAND,IWP/"42142,"42144,"777777700000,
      1 "53640/
        WRITE(5,106)
    106 FORMAT(/'  INPUT,OUTPUT DEVICE NUMBERS?'/)
        READ(5,107) IDEV,IOUT
    107 FORMAT(2I)
        DO 12 I=1,3
     23 READ(IDEV,ERR=12,END=24) IAIR,NDAYS
        GO TO 124
     24 GO TO 23
    124 CONTINUE
        IN=IAIR.AND.IAND
        INAME=IN.OR.ID1
        IN3=IN.OR.IWP
        CALL IFILE(20,INAME)
        CALL IFILE(22,IN3)
        READ(20) IA2
        READ(22,3) IA3
      3 FORMAT(2X,A3)
        WRITE(5,105) NDAYS
    105 FORMAT('  NDAYS ='I5)
        DO 42 IDAY=1,NDAYS
        READ(20) ID,VS,VN,XM
        READ(22,5) WTH
      5 FORMAT(24I2)
        WRITE(IOUT) VS,VN,XM,WTH
        READ(IDEV) IDAY2,TOTAL,VOL,ADR,CONCEN,PMIXP,PF
        DO 40 IH=1,24
        DO 41 IM=1,8
     41 SM(IM)=PMIXP(IH,IM)
        WRITE(IOUT) IH,SM,ADR(IH),CONCEN(IH)
        IS=(IH-1)*60+1
        IFIN=IS+59
        IMIN=0
        ARR=0.
        DEP=0.
        DO 44 INOW=IS,IFIN
      IMIN=IMIN+1
      DO 43 IAD=1,2
   43 VH(IMIN,IAD)=VOL(INOW,IAD)
      ARR=ARR+VOL(INOW,1)
      DEP=DEP+VOL(INOW,2)
   44 CONTINUE
      WRITE(IOUT) ARR,DEP
      WRITE(IOUT) VH
   4Ø CONTINUE
   42 CONTINUE
      CALL RELEAS(2Ø)
      END FILE IOUT
      CALL RELEAS(22)
   12 CONTINUE
      END
```

```
*****************************************************************
*                        INCIDENTAL PROGRAMS
*****************************************************************
* FOLLOWING TEXT PRINTED FROM FILE DSKE:JACK.F4    [4255,516] 18-MAY-76
*****************************************************************
C             JACK.F4
C
      DIMENSION XM(8),SM(8),VH(6Ø,2)
      INTEGER WTH(24)
      WRITE(5,97)
   97 FORMAT('  HOW MANY DAYS DO YOU WANT PRINTED?'/)
      READ(5,98) NDAYS
   98 FORMAT(I)
      DO 5 JDAY=1,NDAYS
      IF(JDAY.GT.1) WRITE(3,6ØØ)
  6ØØ FORMAT(1H1)
      READ(16) VS,VN,XM,WTH
      WRITE(3,1ØØ) VS,VN,XM,WTH
      DO 1 JH=1,24
      READ(16) IH,SM,ADR,CONCEN
      WRITE(3,2ØØ) IH,SM,ADR,CONCEN
      READ(16) ARR,DEP
      WRITE(3,3ØØ) ARR,DEP
      READ(16) VH
      WRITE(3,4ØØ) ((VH(J,1),VH(J,2)),J=1,6Ø)
    1 CONTINUE
    5 CONTINUE
  1ØØ FORMAT(/1ØX,'VS= ',F12.Ø,1ØX,'VN= ',F12.Ø/8G12.4/5X,24I2)
  2ØØ FORMAT(5X,'HOUR ='I3/
     1 8G12.4/
     2 5X,'ADR =',F12.6,5X,'CONCEN =',F12.6)
  3ØØ FORMAT('  ARRIVALS =',F12.6,5X,' DEPARTURES ='F12.6)
  4ØØ FORMAT(5(5X,2F12.6))
      END
```

PART II: USER'S MANUAL FOR APM.SAV

1. INTRODUCTION

The overall objective of this project is the development of an interactive airport performance model, the purpose of which is to translate improvements in airport facilities and airport operating procedures into increases in the quality of service or level of service actually provided to the public.

The program consists of two major segments:

(1) an airside segment which measures delays for the aircraft and passengers in terms of time and dollars,

(2) a groundside segment which determines how much facility is required for peak passenger movements.

The model will use as inputs airport characteristics, processing capabilities, demand patterns, traffic and weather characteristics, and will provide a measure of the change in airport capabilities as reflected by delay time (passengers and aircraft), delay costs (passengers and airlines), serviceable passenger demand (for constant delay) and passenger terminal congestion.

TYPES OF TERMINALS:

Computer Devices (CDI's), Execuports, Digital DECwriters, Hazeltine and Texas Instrument, etc. In effect, those computer terminals that are capable of transmitting at 110, 300, or 1200 baud rates.

TELEPHONE NUMBERS AND BAUD RATES:

110 BAUD AND TTY NUMBER

TEL. #	TTY #	TEL. #	TTY #
2984	6	2986	11
2985	10	2987	12

300 BAUD AND TTY #

TEL. #	TTY #	TEL. #	TTY #
2882	41	2949	27
2883	42	2950	26
2885	54	2951	24
2886	44	2952	21
2888	43	2953	20
2929	50	2954	16
2930	51	2955	13
2931	52	2956	17
2932	53	2957	1
2934	55	2958	5
2935	56	2959	22
		2960	23
		2961	14
		2962	25
		2969	7
		2970	57

1200 BAUD AND TTY #

TEL. #	TTY #
2869	40
2916	46
2917	47

The underlined telephone numbers (2984, 2882 and 2949) are on a rotary system. By dialing 2984, the user will have all of the 110 baud numbers to use. On the 300 baud line, by dialing an underlined telephone number (2882 or 2949), the user will have access to those numbers in the same column.

2. LOGIN

LOGIN is the system program used to gain access to the DECsystem-10. The program determines by appropriate dialogue with the user who he is and whether or not he is currently authorized to use the system. If so, it establishes the user's initial profile, informs him of any messages of the day, whether it is his first logon of the day, and reports any errors detected on his disk files.

Upon dialing any of the appropiate telephone numbers, the word LOG is typed, followed by a↵ (carriage return, RETURN key). After each of the user's responses, the carriage return or RETURN key is to be activated. This indicates the special command has been terminated.

When the symbol # is displayed-insert the user's number.

NAME: is displayed-insert the user's name, which will not be displayed.

PASSWORD: is displayed-insert the user's password which also will not be displayed.

After correctly following this procedure, the messages of the day, if any, are displayed and the program is now ready to run.

If an incorrect NUMBER, NAME or password is entered, the same sequence (except LOG) will be recycled by the system.

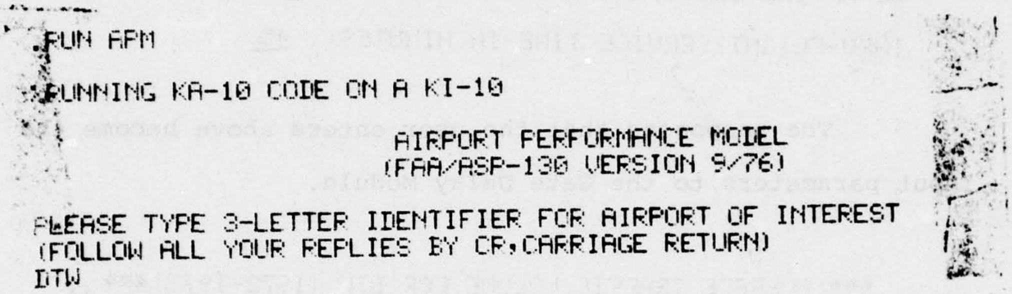
The line

? LGNIET INVALID ENTRY-TRY AGAIN

will be printed a maximum of four (4) times. Then, if an incorrect response is still being entered, the system will print KJOB. The user should check his NUMBER, NAME and PASSWORD. He must then re-enter and the process will begin again.

3. RUNNING THE PROGRAM (AFTER LOGIN)

The following paragraphs are an explanation of, and options to Program APM. After each instruction or message, the RETURN key must be activated. To distinguish between the user commands and the program responses, user entries will be underlined and program responses will not. The program will be loaded and made ready for the user.



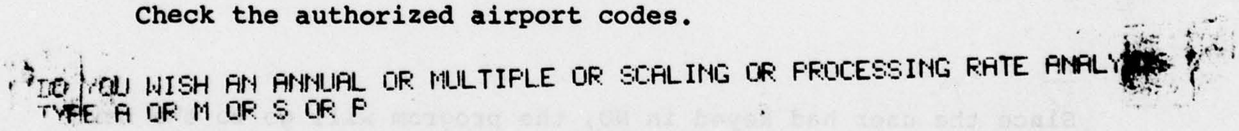
```
.RUN APM

RUNNING KA-10 CODE ON A KI-10

                 AIRPORT PERFORMANCE MODEL
              (FAA/ASP-130 VERSION 9/76)

PLEASE TYPE 3-LETTER IDENTIFIER FOR AIRPORT OF INTEREST
(FOLLOW ALL YOUR REPLIES BY CR,CARRIAGE RETURN)
DTW
```

If a combination of letters other than those included in the list of airports is used, the message

INCORRECT ENTRY.PLEASE TRY AGAIN

is typed by the program.

Check the authorized airport codes.



```
DO YOU WISH AN ANNUAL OR MULTIPLE OR SCALING OR PROCESSING RATE ANALY
TYPE A OR M OR S OR P
```

The user enters his choice. By entering M (multiple analysis), the user can compare differences or similarities in particular parameters for a series of years. An annual analysis, A, is selected when the user desires one particular year. A scaling analysis, S, is for when the user wishes to see the effect of the processing rate upon the T-table. If P is selected, the user will be able to enter a scaling factor and see what effect that scaling factor has upon the processing rate.

```
TO COMPUTE GATE DELAYS:
   TYPE THE # OF SERVICE GATES:  30
    ARRIVAL SERVICE TIME IN MINUTES:  20
     DEPARTURE SERVICE TIME IN MINUTES:  30
     THRU-FLIGHT SERVICE TIME IN MINUTES:  45
```

The responses that the user enters above become the input parameters to the Gate Delay Module.

```
                *** AVERAGE TRAFFIC VOLUME FOR DTW (1972-1973)***

                                         OPERATIONS/DAY      OPERATIONS/YR
AIR CARRIER + COMMUTER/AIR TAXI (AC+CAT)   421                 153843
    GENERAL AVIATION + MILITARY (GA+MIL)   316                 115697

                                     TOTAL   737                 269540

DO YOU WISH TO MODIFY THE ABOVE TOTAL DEMANDS ?
TYPE YES OR NO
NO
```

<u>NO</u>

Since the user had keyed in <u>NO</u>, the program will go to the next section.

If the user had keyed in <u>YES</u> as his response (annual analysis), the program types the following messages and awaits the user's responses.

```
     TYPE TOTAL OPERATIONS FOR THE YEAR (AC+CAT): 200000

     TYPE TOTAL OPERATIONS FOR THE YEAR (GA+MIL): 110000
```

(The user has replied with 200000 and 110000 respectively).

The program then prints the following message for the user so that he may again see the numbers he has entered and make any changes, if necessary.

```
AC+CAT =          200000
GS+MIL =          110000

 TOTAL =          310000
                *** AVERAGE TRAFFIC VOLUME FOR DTW (1972-1973)***
                                               OPERATIONS/DAY     OPERATIONS/YR
AIR CARRIER + COMMUTER/AIR TAXI (AC+CAT)  547                  200000
   GENERAL AVIATION + MILITARY (GA+MIL)   301                  110000

                                    TOTAL  848                  310000

DO YOU WISH TO MODIFY THE ABOVE TOTAL DEMANDS ?
TYPE YES OR NO
```

If the user replies <u>YES</u>, the program will recycle.

If the user replies <u>NO</u>, the program continues.

<u>NO</u>

Since the user had keyed in <u>NO</u>, the program will go to the next section.

If the user selects M for a multiple year analysis, the following section will be entered.

```
MULTIPLE ANALYSIS
  YEAR  1978
```

The user enters 1978, the starting year of his multi-year analysis.

```
WITH OR WITHOUT INVESTMENT ?
    TYPE W OR WO:  WO
```

A data file will now be open for the user. The file will be in the format:

XXXWO.DAT or XXXWI.DAT

The XXX represents the three leter airport code. The WO represents "without investment" and the WI represents "with investment."

The delay cost figures will be written on this file.

```
IS THIS THE INITIALIZATION RUN ?
   YES OR NO:  YES
```

This reply by the user indicates whether or not this is the first run of a multi-year analysis.

```
THE # OF DAYS OR RUNS USED FOR ANNUALIZATION IS PRESENTLY 54.
DO YOU WISH TO CHANGE IT?  (YES OR NO)
NO
```

In this case, 54 is taken from the header file. For various reasons, (checkout, etc.) the user may want a lesser number of days or runs to simulate a year.

If the user had keyed in <u>YES</u> as his response, the program would have typed the following message and waited for the user's reply.

```
NEW NUMBER:     45
```

```
                *** MIX OF AIRCRAFT TYPES AT DTW 1975 ***

 TYPE                                                PERCENT
H4W HEAVY JETS - 4 ENGINE WIDE BODY                    1.8
H3  HEAVY JETS - 3 ENGINE                              3.6
H4S HEAVY JETS - 4 ENGINE STANDARD BODY               14.2
L3  LARGE JETS - 3 ENGINE                             22.6
L2  LARGE JETS - 2 ENGINE                             14.9
LP  LARGE PROP + TURBOPROP                            13.5
S   SMALL (12500 LBS. OR LESS)                        29.4
O   OTHER                                              0.0
                              *** DO YOU WISH TO ***
 1: USE FORECASTED MIX
 2: INSERT A NEW MIX
 3: NEITHER
TYPE 1, 2 OR 3
?1
```

If the user had selected M, for multi-year analysis, the following message would have appeared.

```
TYPE T-TABLE SCALING FACTOR
(0. IF YOU WISH PROGRAM TO COMPUTE IT):    0.
```

The user enters his choice and the program continues.

If S (for scaling analysis) is selected, the program prints the following message and awaits the user's reply.

```
   TYPE PROCESSING RATE FOR COMPUTING SCALE FACTOR :120
```

Here, the user has entered 120. The program will then print the T-table scaling factor for the user; then it will exit.

```
T-TABLE SCALING FACTOR =    0.2085328E+01

CPU TIME: 0.54  ELAPSED TIME: 1:11.05
NO EXECUTION ERRORS DETECTED

EXIT
```

If P (for a processing rate analysis) is selected, the program prints the following message and awaits the user's reply.

```
TYPE SCALING FACTOR FOR COMPUTING PROCESSING RATE: 2.5
```

After the user entered (in this case 2.5), the program will print the scaled processing rate for his entry; then the program will exit.

```
SCALED PROCESSING RATE =  144

CPU TIME: 0.53  ELAPSED TIME: 1:15.72
   EXECUTION ERRORS DETECTED

EXIT
```

With either the annual analysis or the multiple analysis, the program continues with the following section.

```
           *** MIX OF AIRCRAFT TYPES AT DTW 1975 ***
  TYPE                                          PERCENT
H4W HEAVY JETS - 4 ENGINE WIDE BODY                1.8
H3  HEAVY JETS - 3 ENGINE                          3.6
H4S HEAVY JETS - 4 ENGINE STANDARD BODY           14.2
L3  LARGE JETS - 3 ENGINE                         22.6
L2  LARGE JETS - 2 ENGINE                         14.9
LP  LARGE PROP + TURBOPROP                        13.5
S   SMALL (12500 LBS. OR LESS)                    29.4
O   OTHER                                          0.0
                          *** DO YOU WISH TO ***
  1: USE FORECASTED MIX
  2: INSERT A NEW MIX
  3: NEITHER
TYPE 1, 2 OR 3
```

If the user enters the number 1, the following message is typed by the program.

```
TYPE IN ONE OF THE FOLLOWING YEARS
FOR THE FAA PROJECTED MIX TO BE USED:
     1980    1985    1990    1995    2000
```

The user now selects a year, 1995 for example, and the following message is printed.

```
PROJECTED MIX OF AIRCRAFT TYPES AT DTW,1995

                TYPE                     PERCENT

HEAVY JETS                                  9.0
H3 HEAVY JETS - 3 ENGINE                   29.0
L4 LARGE JETS - 4 ENGINE                    0.0
L3 LARGE JETS - 3 ENGINE                   29.0
L2 LARGE JETS - 2 ENGINE                   33.0
LP LARGE PROP + TURBOPROP                   0.0
S SMALL (12500 LBS. OR LESS)                0.0
O OTHER                                     0.0
                              *** DO YOU WISH TO ***
 1: USE FORECASTED MIX
 2: INSERT A NEW MIX
 3: NEITHER
TYPE 1, 2 OR 3
```

If the user enters 2, to insert his own aircraft mix, the program prints a heading and waits for the user to enter his desired percentage of heavy jets, e.g., 2.1. This will be repeated for the seven other categories of aircraft.

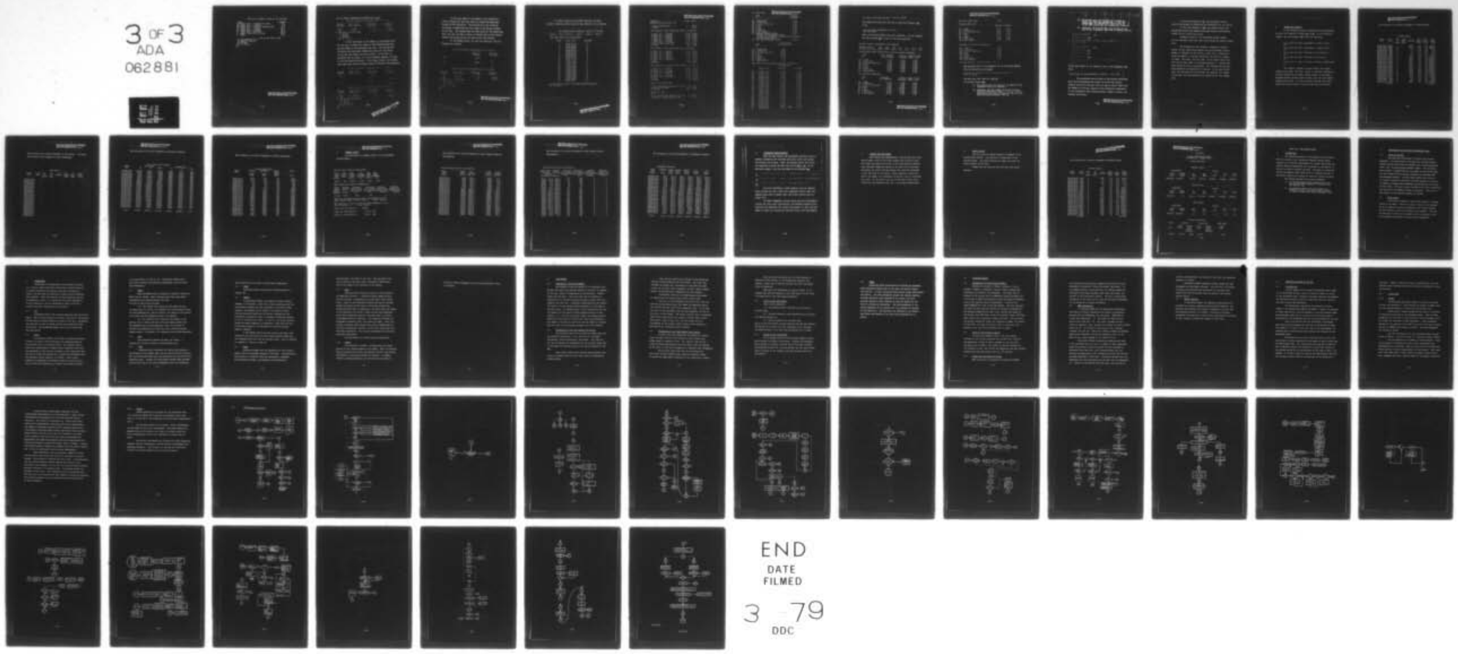
```
MIX OF AIRCRAFT TYPE AT DTW

         TYPE                      PERCENT

H4 HEAVY JETS 4 - ENGINE             2.1

H3 HEAVY JET - 3 ENGINE              4.2

L4 LARGE JETS - 4 ENGINE             11.7

L3 LARGE JETS - 3 ENGINE             21.6

L2 LARGE JETS - 2 ENGINE             17.4

LP LARGE PROP +TURBOPROP             12.3

S SMALL (12500 LBS. OR LESS)         30.7

O OTHER                               0.0
```

The program then prints the aircraft types and percents which the user has just entered so that he may examine his selections and make the necessary corrections, if any. By typing 2, the user causes the program to re-cycle and allow the user to make his changes.

By typing 3, the user's selections are used and the program will continue. In the example below, the user has typed 3; therefore the program continues.

AD-A062 881   TRANSPORTATION SYSTEMS CENTER CAMBRIDGE MASS   F/G 1/5
THE AIRPORT PERFORMANCE MODEL. VOLUME II. USER'S MANUAL AND PRO--ETC(U)
OCT 78   J BELLANTONI
UNCLASSIFIED   TSC-FAA-28-21.2.   FAA-ASP-78-10.2   NL

3 OF 3
ADA
062881



END
DATE
FILMED
3 79
DDC

```
                   *** MIX OF AIRCRAFT TYPES AT DTW 1975 ***

   TYPE                                                    PERCENT
  H4W HEAVY JETS - 4 ENGINE WIDE BODY                         2.1
  H3  HEAVY JETS - 3 ENGINE                                   4.2
  H4S HEAVY JETS - 4 ENGINE STANDARD BODY                    11.7
  L3  LARGE JETS - 3 ENGINE                                  21.6
  L2  LARGE JETS - 2 ENGINE                                  17.4
  LP  LARGE PROP + TURBOPROP                                 12.3
  S   SMALL (12500 LBS. OR LESS)                             30.7
  O   OTHER                                                   0.0

                                       *** DO YOU WISH TO ***
   1: USE FORECASTED MIX
   2: INSERT A NEW MIX
   3: NEITHER
  TYPE 1, 2 OR 3
  3
```

For an annual analysis the program next prints

```
                   *** MAX ACHIEVABLE PROCESSING RATE AT DTW ***

WEATHER      LOWER LIMITS             OPERATIONS            EQUIVALENT
CATEGORY     CLNG FT/VIS MI            PER HOUR               PANCAP

  VFR          1500/3                    75                   233148.
  IFR            0/0                     75                   233148.
DO YOU WISH TO MODIFY ABOVE
    1: OPS/HR
    2: PANCAP
    3: NEITHER ?
TYPE IN 1,2 OR 3
```

If the user types in the number 1, the program waits for the user to change the number of VFR operations per hour and the number of IFR operations per hour. The program then calculates the equivalent PANCAP and prints out the values so the user may examine his input values. If the user is satisfied with his input, as in the following example, he types 3, and the program continues. If he types 1 again, the program will again wait while he enters values for VFR and IFR operations.

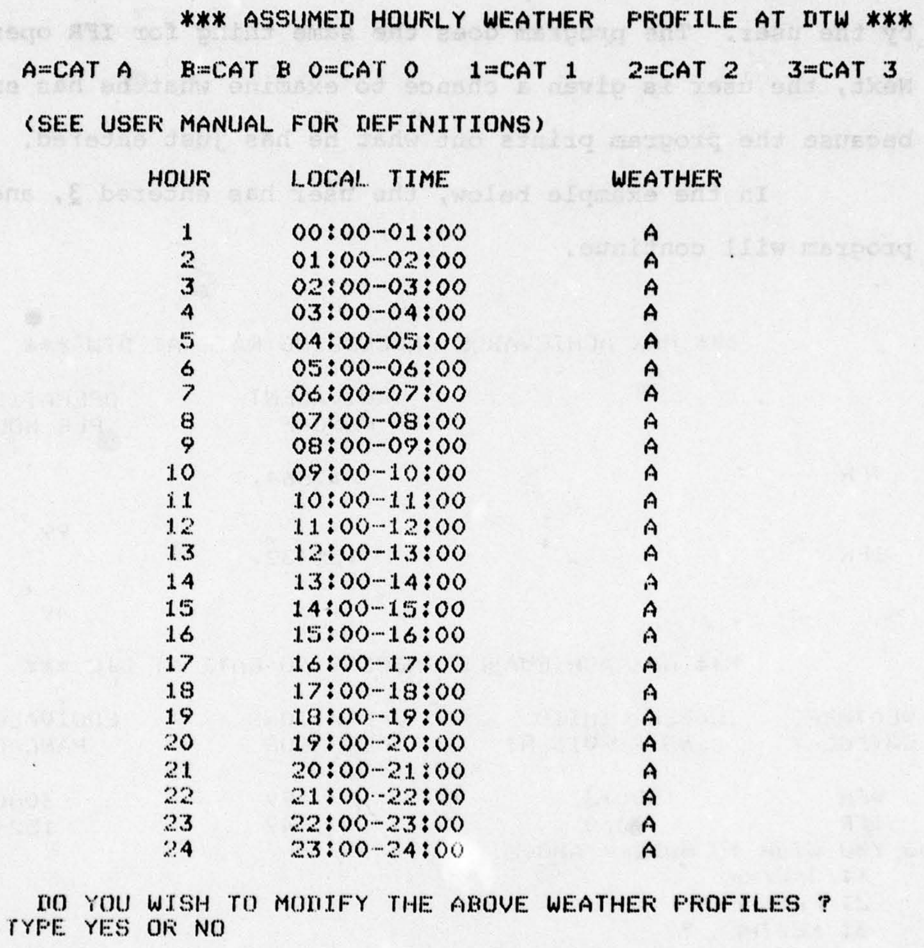
```
                   *** MAX ACHIEVABLE PROCESSING RATE AT DTW ***

WEATHER      LOWER LIMITS             OPERATIONS            EQUIVALENT
CATEGORY     CLNG FT/VIS MI            PER HOUR               PANCAP

  VFR                                    100

                                                              310864.

  IFR                                     50

                                                              155432.

                   *** MAX ACHIEVABLE PROCESSING RATE AT DTW ***

WEATHER      LOWER LIMITS             OPERATIONS            EQUIVALENT
CATEGORY     CLNG FT/VIS MI            PER HOUR               PANCAP

  VFR          1500/3                    100                  310864.
  IFR            0/0                      50                  155432.
DO YOU WISH TO MODIFY ABOVE
    1: OPS/HR
    2: PANCAP
    3: NEITHER ?
TYPE IN 1,2 OR 3
3
```

If the user types in the number 2, the program will print a heading and then wait while he enters the equivalent pancap for VFR operations. The program will then calculate the number of operations per hour based on the pancap entered by the user. The program does the same thing for IFR operations. Next, the user is given a chance to examine what he has entered because the program prints out what he has just entered.

In the example below, the user has entered 3, and the program will continue.

```
           *** MAX ACHIEVABLE PROCESSING RATE AT DTW ***

                              EQUIVALENT          OPERATIONS
                                PANCAP             PER HOUR

  VFR                          310864.

                                                      99
  IFR                          155432.

                                                      49

           *** MAX ACHIEVABLE PROCESSING RATE AT DTW ***

WEATHER      LOWER LIMITS      OPERATIONS          EQUIVALENT
CATEGORY     CLNG FT/VIS MI     PER HOUR             PANCAP

  VFR            1500/3               99             308045.
  IFR              0/0                49             152467.
DO YOU WISH TO MODIFY ABOVE
   1: OPS/HR
   2: PANCAP
   3: NEITHER ?
 TYPE IN 1,2 OR 3
3
```

In another option of the DAILY analysis, the user is given a weather profile which he may change if he so desires.



```
              *** ASSUMED HOURLY WEATHER  PROFILE AT DTW ***

   A=CAT A    B=CAT B O=CAT 0    1=CAT 1    2=CAT 2    3=CAT 3

   (SEE USER MANUAL FOR DEFINITIONS)

            HOUR       LOCAL TIME          WEATHER

              1       00:00-01:00             A
              2       01:00-02:00             A
              3       02:00-03:00             A
              4       03:00-04:00             A
              5       04:00-05:00             A
              6       05:00-06:00             A
              7       06:00-07:00             A
              8       07:00-08:00             A
              9       08:00-09:00             A
             10       09:00-10:00             A
             11       10:00-11:00             A
             12       11:00-12:00             A
             13       12:00-13:00             A
             14       13:00-14:00             A
             15       14:00-15:00             A
             16       15:00-16:00             A
             17       16:00-17:00             A
             18       17:00-18:00             A
             19       18:00-19:00             A
             20       19:00-20:00             A
             21       20:00-21:00             A
             22       21:00-22:00             A
             23       22:00-23:00             A
             24       23:00-24:00             A

   DO YOU WISH TO MODIFY THE ABOVE WEATHER PROFILES ?
  TYPE YES OR NO
```

```
PARAMETER                                    VALUE   UNITS
RADAR APPROACH SPACING STDS.

 1 SMALL BEHIND HEAVY                          6      MILES
 2 OTHER                                       3      MILES

1975 AIRCRAFT DIRECT OPERATING COSTS, IN THE AIR

 3 HEAVY JET - 4 ENGINE                      24.53   $/MIN
 4 HEAVY JET - 3 ENGINE                      17.92   $/MIN
 5 LARGE JET - 4 ENGINE                      13.45   $/MIN
 6 LARGE JET - 3 ENGINE                      10,42   $/MIN
 7 LARGE JET - 2 ENGINE                       8,34   $/MIN
 8 LARGE PROP + TURBOPROP                    14.96   $/MIN
 9 SMALL(12500 LB OR LESS)                    0.38   $/MIN
10 OTHER                                      0.00   $/MIN

1975 AIRCRAFT DIRECT OPERATING COSTS, ON GROUND

11 HEAVY JET - 4 ENGINE                      17,56   $/MIN
12 HEAVY JET - 3 ENGINE                      13,32   $/MIN
13 LARGE JET - 4 ENGINE                      10,12   $/MIN
14 LARGE JET - 3 ENGINE                       8,27   $/MIN
15 LARGE JET - 2 ENGINE                       6,87   $/MIN
16 LARGE PROP + TURBOPROP                    14.13   $/MIN
17 SMALL                                      0,32   $/MIN
18 OTHER                                      0,00   $/MIN

1975 AIRCRAFT POLLUTION EMISSION LEVELS, IN THE AIR

19 HEAVY JET - 4 ENGINE                     320,00   LBS/HR
20 HEAVY JET - 3 ENGINE                     160,00   LBS/HR
21 LARGE JET - 4 ENGINE                      80,00   LBS/HR
22 LARGE JET - 3 ENGINE                      40,00   LBS/HR
23 LARGE JET - 2 ENGINE                      20,00   LBS/HR
24 LARGE PROP + TURBOPROP                    10,00   LBS/HR
25 SMALL                                      5,00   LBS/HR
26 OTHER                                      1,00   LBS/HR

VALUE OF PASSENGER TIME

27 AIR CARRIER AND COMMUTER/AIR TAXI         12.50   $/HR
28 GENERAL AVIATION                          12.50   $/HR
29 MILITARY                                  12.50   $/HR

NUMBER OF GATES

30 AIR CARRIER AND COMMUTER/AIR TAXI         41      GATES
31 GENERAL AVIATION AND MILITARY              5      GATES
```

```
FOR WORK TRIPS

        MODE                                      VEHICLE
         #                                       OCCUPANCY

49      1 DRIVE,PARK                                1.10
50      2 CURB PU+D                                 1.10
90      3 S.T.PK, CURB PU+D                         1.10
51      4 TAXI                                      1.10
52      5 BUS                                      20.00
91      6 LIMO                                      1.10
92      7 RENT CAR                                  1.10
53      8 NON-HIGHWAY                               0.00
54      HIGHWAY PEAKING FACTOR                      1.50
        (AVERAGE FLOW PER MINUTE DURING
         20 MIN PEAK DIVIDED BY AVERAGE FLOW PER MINUTE
         FOR THE ENTIRE HOUR )
55      CURB DESIGN PEAKING FACTOR                  1.50

                              VEHICLE CURB
        MODE                 SLOT LENGHT(FT)
         #

56      1 DRIVE,PARK                0.00
57      2 CURB,PU+D                18.00
58      3 S.T.PK,CURB PU+D         18.00
59      4 TAXI                     18.00
60      5 BUS                      45.00
61      6 LIMO                     25.00
62      7 RENT CAR                  0.00
63      8 NON-HIGHWAY               0.00

WORK FORCE AIRPORT ARRIVAL/DEPARTURE DISTRIBUTION
                              % OF WORKFORCE

           TIME OF DAY              INBOUND               OUTBOUND

64         00:00-01:00               0.000                  0.030
65         01:00-02:00               0.000                  0.010
66         02:00-03:00               0.000                  0.010
67         03:00-04:00               0.010                  0.010
68         04:00-05:00               0.010                  0.000
69         05:00-06:00               0.060                  0.010
70         06:00-07:00               0.280                  0.010
71         07:00-08:00               0.140                  0.050
72         08:00-09:00               0.040                  0.020
73         09:00-10:00               0.030                  0.030
74         10:00-11:00               0.030                  0.030
75         11:00-12:00               0.010                  0.040
76         12:00-13:00               0.040                  0.020
77         13:00-14:00               0.060                  0.010
78         14:00-15:00               0.090                  0.030
79         15:00-16:00               0.030                  0.070
80         16:00-17:00               0.010                  0.250
81         17:00-18:00               0.010                  0.070
82         18:00-19:00               0.020                  0.050
83         19:00-20:00               0.020                  0.040
84         20:00-21:00               0.020                  0.040
85         21:00-22:00               0.030                  0.040
86         22:00-23:00               0.050                  0.050
87         23:00-24:00               0.010                  0.080
```

```
IS THERE A GATE WAITING AREA ? TYPE YES OR NO
```

The program now waits for the user to type his response, YES or NO

```
ARE GATE HOLD PROCEDURES IN EFFECT ?
TYPE YES OR NO
```

Again, the program awaits the user's response. For our example, the user replied YES to both of the above questions.

```
SEATING CAPACITY,CARRIER+AIR TAXI/COMMUTER

AIRCRAFT TYPE =       H4W     H3W     H4S     L3      L2      LP     S/O
32 REV SEATS AVAIL.  346.0   299.0   157.4   115.7   101.1   49.3    6.0

ACCESS/EGRESS MODE CHARACTERISTICS FOR ENPLANING PASSENGERS

     MODE                    WELL WISHERS       VEHICLE   HOURS OF CURB
     #                       PER PASSENGER      OCCUPANCY  DWELL TIME

33   1 PARK                      0.960            1.790      0.000
34   2 CURB PU+D                 2.100            2.500      0.062
35   3 S.T.PK, CURB PU+D         2.100            2.500      0.062
36   4 TAXI                      0.000            1.400      0.033
37   5 BUS                       0.290           20.000      0.066
38   6 LIMO                      0.290            7.000      0.083
39   7 RENT CAR                  0.000            1.790      0.000
40   8 NON-HIGHWAY               0.290            0.000      0.000

FOR DEPLANING PASSENGERS

     MODE                      GREETERS         VEHICLE   HOURS OF CURB
     #                       PER PASSENGER      OCCUPANCY  DWELL TIME

41   1 PARK                      1.030            1.790      0.000
42   2 CURB PU+D                 2.060            2.300      0.062
43   3 S.T.PK, CURB PU+D         2.060            2.300      0.062
44   4 TAXI                      0.000            1.500      0.033
45   5 BUS                       0.140           20.000      0.066
46   6 LIMO                      0.140            8.000      0.083
47   7 RENT CAR                  0.000            1.400      0.000
48   8 NON-HIGHWAY               0.140            0.000      0.000
```

```
88 TOTAL LABOR FORCE                        2857

EMPLOYEE MODE SPLIT
                                        INBOUND  OUTBOUND

93   1 PARK                               0.90      0.90
94   2 CURB PU+D                          0.00      0.00
95   3 S.T.PK, CURB PU+D                  0.00      0.00
96   4 TAXI                               0.00      0.00
97   5 BUS                                0.10      0.10
98   6 LIMO                               0.00      0.00
99   7 RENT CAR                           0.00      0.00
100  8 NON-HIGHWAY                        0.00      0.00

PASSENGER CAR VEHICLE EQUIVALENCY

101  1 PARK                               1.00
102  2 CURB PU+D                          1.00
103  3 S.T.PK, CURB PU+D                  1.00
104  4 TAXI                               1.00
105  5 BUS                                2.50
106  6 LIMO                               1.50
107  7 RENT CAR                           1.00
108  8 NON-HIGHWAY                        1.00

109 LANE CAPACITY (VEHICLES/HOUR)  1800.00
```

The parameter list is now complete and the following message will be printed by the program.

```
DO YOU WISH TO MODIFY ANY OF THE ABOVE PARAMETERS ?
TYPE YES OR NO
```

The user will then type his response.

If the user enters YES:

(1) The program asks the user for the number of the parameter that he is changing.

(2) Depending upon the number of entries in that particular parameter number - 1, 2, or 3 - the program asks the user to enter the new value for that particular parameter section.

(3) The program will ask the user if he wishes to modify any additional parameters. If the user answers YES, the program will recycle with 1). If the user answers NO, the program will continue.

(4) Currently, the program will also continue if the user enters a parameter less than or greater than 88.

```
NUMBER OF PARAMETER =  16
NEW VALUE(S) =  5.30
DO YOU WISH TO MODIFY ANY ADDITIONAL PARAMETERS ?
TYPE YES OR NO
YES
NUMBER OF PARAMETER =  34
NEW VALUE(S) =  2.5
NEW VALUE(S) =  3.1
NEW VALUE(S) =  1.72
DO YOU WISH TO MODIFY ANY ADDITIONAL PARAMETERS ?
TYPE YES OR NO
NO
```

If the user enters no, no changes to any of the parameters take place.

```
DO YOU WANT TO SEE GROUNDSIDE STATISTICS - YES OR NO?
```

The groundside service model of the program determines which facilities associated within and around the airport terminal area will need what kind and type of space, based upon the number of arriving, departing and transferring passengers. It also determines their characteristies, modes of access, and terminal facilities.

If the user response is NO, the groundside service area of the program is avoided and no statistics will be printed.

If the user response is YES, the program enters the groundside portion and computes the peak facility requirements needed for the various facilities.

Messages related to the groundside service module will be explained later when they occur during the normal program flow.

The program will now generate a summary of printed output for the runway delay module and the groundside service module, if desired, in the form of a print file. These print files are named in a certain convention, the same as a person is named. The first, the file name, is the actual name of the file and the last name, the filename extension, indicates with which group the file is associated. The filename and the filename extension are separated by a period. In the case of the print file generated by the program, the computer will assign a file name, and the extension will always be LPT (e.g. QCLSQ. LPT).

4. RUNWAY DELAY MODULE

To the following series of questions from the program, the user will respond by typing YES or NO. In the following example, the user answers YES to all the questions.

```
DO YOU WISH AN HOURLY BREAKDOWN OF TAKEOFF DELAYS ?
YES
DO YOU WISH AN HOURLY BREAKDOWN OF LANDING DELAYS ?
YES
DO YOU WISH AN HOURLY BREAKDOWN OF GATE DELAYS ?
YES
DO YOU WISH AN HOURLY BREAKDOWN OF POLLUTION ?
YES
DO YOU WISH AN HOURLY BREAKDOWN OF ENERGY CONSUMPTION ?
YES
```

For takeoff, landing, and gate delays, the output format is the same. The time, number of takeoffs, landings or aircraft delayed at the gates, maximum number in the queue, maximum delayed, average number delayed, and the total number delayed are printed. For the pollutants emitted, the total pounds and excess pounds of fuel for each hour are printed.

The following is an hourly breakdown of landing delays.

```
                         LANDING DELAYS
   LOCAL       NO OF    MAX.    AVR.      NO.     MAX.    AVR.    TOTAL
    TIME        LAND     IN      IN     DELAYED  DELAY   DELAY    DELAY
                        QUEUE  QUEUE              MINS    MINS     MINS

00:00-01:00  3857.       3                1233     3.0     0.3   1243.7
01:00-02:00  4591.       3                1773     3.1     0.5   2254.7
02:00-03:00  1288.       2                 252     2.2     0.2    305.8
03:00-04:00   229.       0                   0     0.0     0.0      0.0
04:00-05:00   587.       1                  21     1.1     0.0     17.8
05:00-06:00   208.       0                   0     0.0     0.0      0.0
06:00-07:00  5504.       5                2700     5.2     0.9   4904.8
07:00-08:00  2718.       2                 880     1.7     0.3    825.3
08:00-09:0011113.        7                7697     6.8     1.4  15358.9
09:00-10:00  9151.       7                6454     7.1     1.6  15008.0
10:00-11:00  6982.       8                4271     9.0     1.7  11955.1
11:00-12:00  8810.       5                5532     4.4     0.9   7685.4
12:00-13:00  8379.       8                5980     8.9     1.6  13824.6
13:00-14:00  9802.       5                6580     5.8     1.1  11181.6
14:00-15:0011237.        7                7910     7.9     1.7  18543.4
15:00-16:00  8663.       4                5432     4.1     0.8   6821.8
16:00-17:0011891.       11                9028    12.5     2.3  27672.9
17:00-18:00  7682.      11                5304    14.0     1.9  14632.1
18:00-19:00  7774.       5                4441     5.8     0.7   5645.7
19:00-20:00  1621.       2                 204     1.7     0.1    155.7
20:00-21:00  5710.       8                2979     8.2     1.7   9766.7
21:00-22:00   148.       1                  88     0.7     0.3     43.5
22:00-23:00  3861.       6                1847     8.1     1.1   4313.4
23:00-24:00  5326.       3                1868     4.1     0.4   1919.9
           137130.                                              174080.8
```

The following is an hourly breakdown of gate delays. Currently, this section of the program is under development.

GATE DELAYS

LOCAL	NO OF	MAX	AV	NO	MAX	AV	TOTAL
TIME	ARRS	IN	IN	DELAYED	DELAY	DELAY	DELAY
		QUEUE	QUEUE		MINS	MINS	MINS
00:00-01:00							
01:00-02:00							
02:00-03:00							
03:00-04:00							
04:00-05:00							
05:00-06:00							
06:00-07:00							
07:00-08:00							
08:00-09:00							
09:00-10:00							
10:00-11:00							
11:00-12:00							
12:00-13:00							
13:00-14:00							
14:00-15:00							
15:00-16:00							
16:00-17:00							
17:00-18:00							
18:00-19:00							
19:00-20:00							
20:00-21:00							
21:00-22:00							
22:00-23:00							
23:00-24:00							

The following is an hourly breakdown of pollutants emitted.

TOTAL POLLUTANTS EMITTED

LOCAL	TOTAL TONS			EXCESS TONS		
TIME	HC	CU	NO(X)	HC	CU	NO(X)
00:00-01:00	28386.	59377.	29608.	4352.	8019.	398.
01:00-02:00	16739.	37898.	11752.	4352.	8019.	398.
02:00-03:00	8680.	17961.	4959.	4352.	8019.	398.
03:00-04:00	8100.	15346.	5706.	4352.	8019.	398.
04:00-05:00	6717.	13350.	3034.	4352.	8019.	398.
05:00-06:00	6062.	11701.	2461.	4352.	8019.	398.
06:00-07:00	20611.	46447.	16217.	4352.	8019.	398.
07:00-08:00	35922.	71453.	43470.	4352.	8019.	398.
08:00-09:00	52542.	114889.	54821.	4352.	8019.	398.
09:00-10:00	57738.	121987.	65724.	4352.	8019.	398.
10:00-11:00	52428.	108834.	61026.	4352.	8019.	398.
11:00-12:00	61407.	128519.	72129.	4352.	8019.	398.
12:00-13:00	57968.	121523.	66905.	4352.	8019.	398.
13:00-14:00	52210.	112587.	56331.	4352.	8019.	398.
14:00-15:00	64367.	137534.	72275.	4352.	8019.	398.
15:00-16:00	65285.	135719.	77714.	4352.	8019.	398.
16:00-17:00	66444.	142333.	74075.	4352.	8019.	398.
17:00-18:00	65398.	134438.	78944.	4352.	8019.	398.
18:00-19:00	60131.	124629.	71360.	4352.	8019.	398.
19:00-20:00	36921.	71996.	46355.	4352.	8019.	398.
20:00-21:00	28449.	61564.	27516.	4352.	8019.	398.
21:00-22:00	30840.	58426.	39625.	4352.	8019.	398.
22:00-23:00	24989.	52654.	24999.	4352.	8019.	398.
23:00-24:00	23309.	51591.	20082.	4352.	8019.	398.
TOTAL	931642.	1952759.	1027087.	104439.	192445.	9554.

The following is an hourly breakdown of energy consumption.

	ENERGY CONSUMPTION			
LOCAL TIME	ARRIVAL	DEPARTURE	GATE DELAY	TOTAL
	TOTAL TONS	TOTAL TONS	TOTAL TONS	TONS
00:00-01:00	1095.	1377.	58.	2530.
01:00-02:00	1227.	187.	58.	1472.
02:00-03:00	318.	145.	58.	521.
03:00-04:00	43.	321.	58.	422.
04:00-05:00	146.	99.	58.	304.
05:00-06:00	78.	95.	58.	231.
06:00-07:00	1559.	370.	58.	1987.
07:00-08:00	684.	2457.	58.	3199.
08:00-09:00	3442.	2171.	58.	5672.
09:00-10:00	2948.	3109.	58.	6115.
10:00-11:00	2281.	3056.	58.	5396.
11:00-12:00	2520.	3547.	58.	6126.
12:00-13:00	2763.	3264.	58.	6086.
13:00-14:00	2962.	2418.	58.	5439.
14:00-15:00	3619.	3270.	58.	6947.
15:00-16:00	2474.	3922.	58.	6454.
16:00-17:00	4222.	3316.	58.	7597.
17:00-18:00	2612.	4129.	58.	6800.
18:00-19:00	2196.	3621.	58.	5876.
19:00-20:00	423.	2755.	58.	3237.
20:00-21:00	1839.	1068.	58.	2965.
21:00-22:00	55.	2497.	58.	2611.
22:00-23:00	1180.	1113.	58.	2352.
23:00-24:00	1438.	612.	58.	2108.

5. SUMMARY OUTPUT

The following is a summary output of the groundside service module.

```
 PEAK TERMINAL FACILITY REQUIREMENTS

  MAIN     MAIN     PAX        PAX      BAG     BAG
 LOBBY   LOBBY   COUNTER   COUNTER   CLAIM   CLAIM
  AREA    SEATS     AREA    FRONTAGE   AREA   FRONTAGE
(SQ FT)          (SQ FT)     (FT)    (SQ FT)    (FT)

  68042     3719   7323.8     183.1   31150      890

 PEAK HIGHWAY AND PARKING REQUIREMENTS

 SHORT    EMPLOYEE    ENPLANING       DEPLANING      INBOUND        OUTBOUND
  TERM    PARKING    CURB LENGTH    CURB LENGTH ACCESS ROAD    ACCESS ROAD
PARKING              REQUIREMENTS  REQUIREMENTS REQUIPEMENTS REQUIREMENTS
(SLOTS)   (SLOTS)       (FT)           (FT)          (LANES)        (LANES)

  199.2   7639.2        1021             996               5              5

 PEAK HOUR LONG-TERM PARKING SPACE REQUIREMENTS OVER AND
 ABOVE SPACES TAKEN AT START OF DAY :       989.2  SPACES

 NET INCREASE IN LONG-TERM PARKING SPACES OCCUPIED AT THE
 END OF THE DAY :           583.9  SPACES

 TOTAL AIRPORT EMPLOYEES :       19054.7

 PEAK HOUR ENPLANEMENTS :            3685.5    PAX

 PEAK HOUR DEPLANEMENTS :            3340.7    PAX
```

The following is an hourly breakdown of peak terminal facility requirements.

PEAK TERMINAL FACILITY REQUIREMENTS

LOCAL TIME (HOUR)	LOBBY AREA (SQ FT)	PAX COUNTER (SQ FT)	BAGG CLAIM (SQ FT)	TIX COUNTER (LN FT)
00:00-01:00	23078	2405.8	14805	60.1
01:00-02:00	5924	528.0	14385	13.2
02:00-03:00	5168	445.7	9030	11.1
03:00-04:00	4880	414.3	5915	10.4
04:00-05:00	4358	357.1	6825	8.9
05:00-06:00	5168	445.7	6265	11.1
06:00-07:00	5114	439.9	12425	11.0
07:00-08:00	26588	2788.8	9205	69.7
08:00-09:00	22016	2288.4	20090	57.2
09:00-10:00	57998	6225.4	31150	155.6
10:00-11:00	51248	5486.7	23975	137.2
11:00-12:00	60950	6546.7	19880	163.7
12:00-13:00	51914	5559.2	28945	139.0
13:00-14:00	47738	5102.3	29225	127.6
14:00-15:00	47198	5044.1	28000	126.1
15:00-16:00	68042	7323.8	24710	183.1
16:00-17:00	43382	4626.3	22610	115.7
17:00-18:00	62930	6763.9	21385	169.1
18:00-19:00	50438	5397.5	18900	134.9
19:00-20:00	48710	5207.8	13090	130.2
20:00-21:00	14294	1444.9	14175	36.1
21:00-22:00	42410	4519.3	5810	113.0
22:00-23:00	36560	3879.2	19775	97.0
23:00-24:00	18884	1947.0	24255	48.7

The following is an hourly breakdown of peak access facility requirements.

PEAK ACCESS FACILITY REQUIREMENTS

LOCAL	SHORT	EMPLOYEE	ENPLANING	DEPLANING	INBOUND	OUTBOUND
TIME	TERM	PARKING	CURB LENGTH	CURB LENGTH	ACCESS ROAD	ACCESS ROAD
	PARKING		REQUIREMENTS	REQUIREMENTS	REQUIREMENTS	REQUIREMENTS
(HOUR)	(SLOTS)	(SLOTS)	(FT)	(FT)	(LANES)	(LANES)
01:00	69.9	-227.1	201	499	1	2
02:00	32.2	-302.7	41	378	1	1
03:00	17.6	-378.4	61	186	1	1
04:00	8.6	-378.4	45	40	1	1
05:00	7.2	-302.7	37	54	1	1
06:00	10.5	155.9	72	41	1	1
07:00	40.4	4365.3	272	227	5	1
08:00	71.8	5768.4	414	198	3	2
09:00	108.0	6080.2	675	499	2	2
10:00	199.2	6080.2	924	957	3	3
11:00	172.5	6080.2	989	747	3	2
12:00	179.8	5612.5	907	607	2	2
13:00	164.6	5924.3	841	893	2	2
14:00	154.4	6703.8	728	996	3	2
15:00	184.7	7639.2	1021	863	4	3
16:00	196.3	7015.6	825	795	2	3
17:00	163.0	3274.0	940	739	2	5
18:00	167.4	2338.5	865	687	2	3
19:00	149.5	1870.8	751	563	2	2
20:00	121.5	1559.0	492	391	2	2
21:00	70.5	1247.2	498	364	2	2
22:00	89.5	1091.3	537	98	2	1
23:00	83.9	1091.3	380	456	2	2
24:00	79.2	0.0	358	647	1	3

The following is an hourly breakdown of passenger movements.

PASSENGER MOVEMENTS

LOCAL TIME (HOUR)	THROUGH (PAX)	TRANSFERS (PAX)	ORIGINATING (PAX)	TERMINATING (PAX)	EN-PLANED (PAX)	DE-PLANED (PAX)
00:00-01:00	65.5	386.0	835.4	859.1	1210.7	1245.1
01:00-02:00	62.6	368.7	183.3	820.7	265.7	1189.4
02:00-03:00	26.5	155.9	154.8	347.1	224.3	503.0
03:00-04:00	5.5	32.3	143.9	71.9	208.5	104.2
04:00-05:00	11.8	69.5	124.0	154.8	179.7	224.3
05:00-06:00	7.9	46.4	154.8	103.2	224.3	149.5
06:00-07:00	49.5	291.7	152.8	649.2	221.4	940.9
07:00-08:00	27.7	163.1	968.3	363.1	1403.4	526.3
08:00-09:00	101.1	595.5	794.6	1325.4	1151.6	1920.8
09:00-10:00	175.8	1035.6	2161.6	2305.1	3132.7	3340.7
10:00-11:00	127.3	749.9	1905.1	1669.0	2761.0	2418.9
11:00-12:00	99.7	587.2	2273.1	1307.1	3294.4	1894.3
12:00-13:00	161.0	948.1	1930.3	2110.2	2797.5	3058.3
13:00-14:00	162.8	959.1	1771.6	2134.8	2567.6	3093.9
14:00-15:00	154.5	910.1	1751.4	2025.6	2538.3	2935.7
15:00-16:00	132.5	780.2	2543.0	1736.7	3685.5	2516.9
16:00-17:00	118.1	695.6	1606.4	1548.3	2328.0	2243.9
17:00-18:00	110.0	647.8	2348.6	1441.8	3403.7	2089.6
18:00-19:00	93.1	548.6	1874.1	1221.1	2716.1	1769.8
19:00-20:00	54.0	317.9	1808.3	707.6	2620.7	1025.5
20:00-21:00	61.2	360.7	501.7	802.9	727.1	1163.6
21:00-22:00	4.8	28.2	1569.2	62.8	2274.2	91.0
22:00-23:00	99.2	584.2	1346.9	1300.2	1952.1	1884.4
23:00-24:00	129.2	761.0	676.0	1693.7	979.8	2454.7
TOTAL	2041.3	12023.2	29579.0	26761.4	42868.1	38784.6

6. **GROUNDSIDE SERVICE MODULE**

When the user selects the groundside statistics option, summary information for selected facilities within and around the terminal are printed. Then, the program prints the following questions to which the user will key in YES or NO. In the following example, the user has keyed in the response YES.

```
DO YOU WISH AN HOURLY BREAKDOWN OF PEAK TERMINAL FACILITY REQUIREMENTS
YES

DO YOU WISH AN HOURLY BREAKDOWN OF PEAK ACCESS FACILITY REQUIREMENTS ?
YES

DO YOU WISH AN HOURLY BREAKDOWN OF PASSENGER MOVEMENTS ?
YES
```

An hourly breakdown of peak terminal facility requirements include the time, lobby area, passenger counter area, and baggage claim area in square feet; and ticket counter area in linear feet.

An hourly breakdown of peak access facility requirements include the time, short term parking, and employee parking slots; enplaning and deplaning curb length requirements in feet; and the number of lanes for inbound and outbound access road requirements.

7. LOGGING OFF THE SYSTEM

When leaving the DECsystem-10, the user will be in the monitor mode (a dot in the first column) and will key in K/F. This saves the user's disk files in the state which he desires them, enforces logged-out quotas on all disk file structures, terminates the user's job and returns the resources allocated to the user back to the system. These resources include the user's job number, his allocated I/O devices, and his allocated core area. At this time, too, all of the user's print files (those with the extension LPT) will be printed automatically.

8. SAMPLE OUTPUT

The first page of sample output is a summary of the airside delay module. This section is always part of the printed output. All other sections of output are under the direct control of the user.

Formats are the same for both the daily and annual analysis.

The following is an hourly breakdown of takeoff delays.

TAKEOFF DELAYS

LOCAL TIME	NO OF T-OS	MAX IN QUEUE	AV IN QUEUE	NO DELAYED	MAX DELAY MINS	AV DELAY MINS	TOTAL DELAY MINS
00:00-01:00	3910.	2		1359	1.6	0.2	841.8
01:00-02:00	615.	1		172	0.9	0.2	108.5
02:00-03:00	199.	1		5	0.4	0.0	2.1
03:00-04:00	939.	1		202	1.1	0.2	177.2
04:00-05:00	269.	1		0	1.1	0.0	0.5
05:00-06:00	339.	0		0	0.0	0.0	0.0
06:00-07:00	942.	2		597	1.5	0.5	480.0
07:00-08:00	7093.	4		3338	3.1	0.5	3748.2
08:00-09:00	6272.	4		3771	3.5	0.5	3093.9
09:00-10:00	9061.	5		6011	4.7	0.8	7679.4
10:00-11:00	8884.	5		4752	3.0	0.5	4579.4
11:00-12:00	9532.	11		6717	7.4	1.3	12014.1
12:00-13:00	10263.	11		7178	11.3	1.5	15802.7
13:00-14:00	7130.	5		4609	3.9	0.7	4829.4
14:00-15:00	9576.	5		6386	4.6	0.8	7211.3
15:00-16:00	11307.	6		7540	4.8	0.9	9817.8
16:00-17:00	9817.	6		6700	5.4	0.7	7123.6
17:00-18:00	12110.	7		7612	6.8	0.8	9281.8
18:00-19:00	10749.	5		7426	3.9	0.8	8137.2
19:00-20:00	8097.	4		2389	3.2	0.3	2252.6
20:00-21:00	3093.	4		1179	2.9	0.3	902.2
21:00-22:00	7369.	7		2798	5.7	0.7	5248.2
22:00-23:00	3262.	3		1290	2.5	0.3	1128.8
23:00-24:00	1765.	2		536	1.5	0.2	312.3
	142595.						104772.9

```
                                12-Aug-76

                        AIRPORT PERFORMANCE MODEL
                      (FAA/ASP-130, VERSION 5/76)

                            ANNUAL ANALYSIS


                             TAKEOFF QUEUE

        POLLUTANTS                                     DELAYS

      TOTAL     EXCESS          AC          PAX          AC          PAX        TOTAL
      TONS      TONS            HRS         HRS          $           $            $

   25944356. 1903777.         1746.2     76988.8     513966.     962360.    1476326.


                             LANDING QUEUE

        POLLUTANTS                                     DELAYS

      TOTAL     EXCESS          AC          PAX          AC          PAX        TOTAL
      TONS      TONS            HRS         HRS          $           $            $

   14761503.       0.         2901.3    127398.3    1240486.    1592479.    2832966.


                              GATE DELAYS

        POLLUTANTS                                     DELAYS

      TOTAL     EXCESS          AC          PAX          AC          PAX        TOTAL
      TONS      TONS            HRS         HRS          $           $            $

    1903777.  1903777.          0.0          0.0          0.          0.          0.


                           ENERGY CONSUMPTION

      TOTAL          ARRIVAL               DEPARTURE             GATE DELAY

                  TOTAL      EXCESS      TOTAL      EXCESS          TOTAL
      TONS        TONS       TONS        TONS       TONS            TONS
                             TYPE 1                 TYPE 2
                            (DELAYS)               (DELAYS)

      100391.     42124.      7945.      48918.         0.           1403.
```

PART III: APM SYSTEM REPORT

1. INTRODUCTION

The overall objective of this project is the development of an easy to use interactive computer package whose purpose is to translate improvements in airport facilities and airport operating procedures into increases in the quality of service or level of service provided. The model is called the Airport Performance Model (APM) since it predicts how well a particular airport will perform in regard to certain criteria.

The program consists of two major segments:

(1) an airside segment which measures delays for the aircraft and passengers in terms of time and dollars; and

(2) a groundside segment which determines the dimensions which a particular facility needs in order to effectively handle passenger flow.

2. DESCRIPTION OF THE AIRPORT PERFORMANCE MODEL

2.1 Input To The APM

The data base information is airport and aircraft dependent. For each airport, the data base contains 24-hour demand profiles which give the weather, mix, arrival-departure ratio, and a minute-by-minute demand profile. The data base also contains a header file for the number of VFR and IFR days, and the volumes for the various categories of scheduled and non-scheduled traffic. A static input header file (for all 31 airports treated by the model) has been created to minimize the amount of primary data gathering required during the course of this study. The user is given the ability to alter selectively the input data at different stages of execution. The data is then passed to the different modules as needed.

2.2 Model Output

A statistical summary of the airside segment is always output by the model. The user is given a choice as to whether or not he desires an hourly breakdown of the airside segment and the statistics which go to make-up that segment. The user is also given a choice as to whether or not he desires to see the groundside statistics and their hourly breakdowns.

3. SUBROUTINES

Subroutines (or subprograms) serve useful functions, i.e., help to reduce duplicate coding areas, because they may be called throughout the program, they may be checked out easily, and they have the potential of being used by different main programs. Since the function of each subroutine may not be immediately clear, it is hoped that the brief explanation of each subroutine, and the anagrammatic names of the subroutines and their variables will be helpful.

3.1 APM

Subroutines APM is the starting subroutine for the entire model. Here, the user selects an airport, then data is initialized and the type of analysis desired is input by the user. Parameters that the user is able to change include the volumes and mixes of aircraft. The groundside segment can also be called from this subroutine.

3.2 PROCS

In subroutine PROCS, the T-table (a minimum acceptance interval) is read in and calculations are performed based on the aircraft mix in a particular hour. The resultant matrix takes into consideration a complete day's operation based upon the percentage of each type of aircraft in that particular hour. The user is given the opportunity to change the processing rate, the practical annual capacity, or neither. When a daily analysis is undertaken, the weather profile is output and the user is given the opportunity to change the weather condition in as many hours, or none at all. Subroutines PARAM, which has various airside and groundside parameters, is called from this subroutine.

3.3 ANULY

When the ANNUAL type of analysis is desired, subroutine ANULY will be called. Here, the particular data files which characterize the selected airport are read.

Two sets of data files are read:

(1) the "annual file" which has the number of runs or passes that will occur, the scheduled and non-scheduled volume, the non-scheduled mix, type of weather, the number of the profile to be used, and the weighting factor to be applied.

(2) the "profile file" which has the number of the profile (one to three) and both a 24-hour demand profile with the scheduled mix, arrival-departure ratio, the percent of arrivals and departures for that hour; and a minute-by-minute demand profile, in percents, for scheduled arrivals and departures.

3.4 HJE1

This subroutine removes the heavy jet effect. A nominal mix is used to effect a new processing rate.

3.5 PTOC

Subroutine PTOC calculates the finite delay capacity. The maximum arrival demand, MAD, and the maximum arrival capacity, MAC, are calculated based upon the processing rate and the arrival-departure ratio. Testing the relationship between AMAC and AMAD, and also the size of the arrival-departure ratio will determine which equation will be used in finite delay computation.

3.6 MIXNJ

Subroutine MIXNJ calculates mix numbers based on a nominal mix.

3.7 PANCAP

In subroutine PANCAP, the Practical Annual Capacity (PANCAP) is caluclated. After the airside processing rate for each weather category has been converted to the finite delay capacity, the peaking factor (which is a percent of daily activity during the peak hour of the day, averaged for the two busiest consecutive hours) is checked to determine whether or not it is within set limits. A ratio of the hourly capacity versus the annual capacity is calculated. That ratio and the finite delay capacity will determine the Practical Annual Capacity (PANCAP).

If the PANCAP figure has been input by the user, this subroutine will calculate the finite delay capacity based upon the aforementioned ratio and the peaking factor. See AC 150/5060-1A, Appendix 2, Figure 9, page 28.

3.8 PARAM

Subroutine PARAM lists the parameters for the airside segment and the groundside segment of the model. Characteristics which pertain to aircraft costs and consumption, passenger arrival-departure modes, and employee arrival-departure distributions, are shown to the user. The subroutine list will be given to the user, with a minimum of interaction, and it will apply to all airports in the system.

3.9 QDLY

Subroutine QDLY calculates delay for both arriving and departing aircraft. A minute-by-minute demand profile for the aircraft is processed on a first-come, first-served basis. Delay is the difference between the time an aircraft enters the system and the time it would have entered the system had there been no congestion. Aircraft enter the system at the beginning of a minute. When arrivals and departures enter their respective queues at the same time, an arrival is processed before a departure. The remaining aircraft are processed alternately until the queues are empty. There is a minimal acceptance time (between the various combinations of arriving and departing aircraft in each hour) which goes into the delay computation. This minimal acceptance time has been previously calculated.

The gate module is called from this subroutine.

3.10 AIROUT

This subroutine, AIROUT, is essentially the output portion of the airside segment of the model. Here, the desired delay statistics for arrival and takeoff aircraft, gate delays, and amounts of pollutants emitted are printed. A summary listing is always generated, but the user has an option of seeing an hourly breakdown of each of the statistical areas of interest.

4. COST MODULE

4.1 Description of the Cost Module

The purpose of the Cost Module is to translate, into dollars and cents, any aircraft delays (on arrival, on takeoff, at the gate, and/or to passengers on the aircraft). Delays waste time and money. The extra time that an aircraft is delayed will run up direct operating costs in extra fuel consumption and crew cost. Passenger dissatisfaction with delays can lead to a lower average seating capacity in each type of aircraft class. An important part of the calculations for the Cost Module are the hourly traffic characteristics at each airport, i.e., the mix of aircraft types, the average seating capacity of each type of aircraft mix, gate hold procedures, the number of aircraft in each hour and the number of aircraft that are able to be processed (perhaps due to the weather).

4.2 Assumptions of the Cost Module (Air Delay)

The operating costs of air delay are broken down into the following categories: crew, fuel and oil, airframe maintenance, engine maintenance, and burden. The costs for the five categories are subdivided for the aircraft types used in the model and similarly used in the various modules of the APM.

Crew costs increase with airborne delay because crew costs are commonly paid by block time, which is affected by airborne delay.

Fuel and oil costs are an average of the ground and airborne costs divided by the total hours. The total cost includes the higher fuel consumption of take-off and climb, as well as the lower fuel consumption of taxi, which together bracket the desired fuel consumption rate for holding.

Maintenance costs are included because maintenance is usually based on engine hours and/or flying hours.

The cost of lost passenger time due to air, gate and take-off delays depends upon the number of passengers aboard the aircraft and the value of the passenger time. In the APM, the value of the passenger time has been assumed to be $12.50 per hour, on the average. The actual average value of passenger time varies widely with trip purpose, aircraft type, origin, destination, time of day, etc., but the figure ($12.50) corresponds to the current (1976) nominal value employed in FAA and DOT air transport benefit studies.

4.3 Assumptions of the Cost Module (Gate Delays)

Aircraft that have landed and are delayed in docking incur the same costs as those waiting to land, except that engine speed is reduced to idle. As a result, fuel and oil costs and engine maintenance costs are lower; both are reduced by the ratio of idle fuel consumption to approach fuel consumption.

Gate arrival delay costs are not calculated for small type aircraft because these vehicles generally unload on aprons and ramps where congestion is not ordinarily a factor.

When gate hold procedures are in effect because of departure runway delays, it is assumed that aircraft will sustain no more than 10 minutes of delay (at full idle power) prior to departure.

When gate hold procedures are not in effect, it is assumed that aircraft operate at full idle power for the first 10 minutes and at half idle power thereafter.

4.4 Input to the Cost Module

Input tables represent:

(1) the operating costs affected by air delay by aircraft type,

(2) aircraft operating costs affected by gate arrival and takeoff delays, and

(3) seating capacity for aircraft type.

Input to the Cost Module includes aircraft mix and load factors, and delays for aircraft arriving, departing and at the gate, including whether or not gate hold procedures are in effect.

4.5 Output of the Cost Module

In the summary portion of the airside segment output, the following statistics are printed: aircraft and passenger hours delayed in landing, take-off and at the gate; aircraft and passenger dollars due to those delays; and the total dollar value of delays to the landing, take-off and gate portion of the program.

4.6 DCOST

Subroutine DCOST calculates the aircraft and passenger delay statistics which occur at the gate, and during landing and take-off. It also calculates their dollar values individually and totally. The data statements contain the seating capacity, aircraft operating costs affected by air delay, and total aircraft operating costs affected by take-off delays for the first 10 minutes and thereafter. The time parameters in the calling sequence are in seconds, but are converted to minutes in the subroutine. The equations are affected by the airport load factor and whether or not gate hold procedures are in effect.

5. POLLUTION MODULE

5.1 Description of the Pollution Module

Whether an aircraft is idling, taxiing, or is at liftoff, climbout or approach, energy is being consumed and pollutants are entering the atmosphere. The prime area of concern is the airport and its immediate vicinity because this is the area in which the pollution level and energy consumption will be the greatest (most concentrated). A static file named ENGEM.DAT is read in the Cost Module. The file contains: the time for each type of aircraft for arrival and departure, for energy consumed during idle, taxi, liftoff and ascent to and descent from 3000 feet; as well as emission levels (10^{-3}/lbs-fuel) during idle, approach, climbout and taxi. The pollution and energy consumption levels are affected by number of aircraft in each hour, the type of aircraft (mix) in each hour, and whether or not gate hold procedures are in effect.

5.2 Input to the Pollution Module

The data file ENGEM is read in the Cost Module. Included in the calling sequences are aircraft mix, arrivals and departures, amount of delay, whether or not gate hold procedures are in effect, the number of aircraft delayed ten (10) minutes or more, and arrival and departure sums for aircraft delayed less than and greater than ten (10) minutes.

5.3 Output for the Pollution Module

When the return is generated to subroutine AIROUT, the calculations for arrival, departure, gate and an hourly breakdown of pollutants (total and excess) take place. In the summary portion of the printout, the various amounts of pollutants and energy consumed, both total and excess, are printed in tons and dollar amounts. The user is then given the option to see either one or both of the hourly breakdowns of the pollution or energy consumption.

5.4 EMIS (Pollution)

In subroutine EMIS and its associated subroutines, the amount of energy consumed and the amount of pollutants emitted into the atmosphere are calculated. ENGEM.DAT is a data file containing: (1) the times of ascent to and descent from 3000 feet; (2) the energy consumed during a minute of approach, idle, takeoff and climbout for each aircraft type; and (3) the emission levels during idle, approach climbout and take off. EMIS calls INITIAL, POLENG and DEPCALC. Subroutines POLLUTION, INITIAL and POLENG are part of the EMIS.F4 file, while DEPCALC is the DEPRT.F4 file.

Subroutine INITIAL calculates variables which need to be calculated once for use in a number of other equations. These calculations include the amount of fuel consumed in descending from 3000 feet, fuel consumed during taxi for arrivals and departures, fuel consumed at liftoff and the fuel consumed ascending to 3000 feet. Pollutants emitted into the atmosphere are also calculated for the same kind of parameters, i.e., ascent to and descent from 3000 feet, taxi period for arrivals and departures, one minute of idle time, and emissions produced at liftoff.

Subroutine POLENG calculates arrival values for fuel consumption and pollutants emitted. For each hour, the values of energy and pollutants are calculated, at the gate, for delays less than and greater than ten minutes, taxi delays and air delays.

5.5 DEPRT (DEPCALC)

In subroutine DEPRT, the departure calculations are performed for energy consumption and pollutants emitted. The pollution calculations will depend on whether or not the gate hold procedure option is in effect. Pollution and energy consumption for each hour of the day are calculated for aircraft at idle, taxiing, liftoff and climbout to 3000 feet.

6. GROUNDSIDE PORTION OF THE APM

6.1 Introduction

A major segment of the Airport Performance Model (APM) is the groundside section, which is actually a model in itself. In the groundside model, terminal and access facility requirements are calculated through a series of equations that are driven by aircraft arrivals and departures.

The groundside calculations and printouts are performed by two subroutines; GNDCAL and GNDOUT. Control is switched to these routines by a calling sequence in the main program (APM). The groundside model can be utilized for either daily runs or annualization runs depending upon the control asserted by the user.

Daily runs dictate that the groundside calculations and printouts are performed for each day in which data is input. For example, if 30 days of data are used in the APM program, 30 separate days of calculations and printouts are computed by the groundside model.

Annualization runs differ from daily runs in that only one printout results regardless of the number of days of data that the APM program uses. The calculations for each day of data are still computed, however the contents of the printout consist of the maximum values of all calculations for all days of data. For example, if 30 days of data are used by the APM program, each day will produce results from the calculations of the equations in the model. However, from the 30 sets of calculations, only the maximum values ascertained over the 30 days are combined to yield one printout.

6.2 Subroutines

6.2.1 GNDCAL

As previously mentioned, GNDCAL is driven by aircraft arrivals and departures. These figures are passed to GNDCAL from the main program where it is input from an external file.

Additionally, several parameters are used in the system of equations which make up the groundside model. Some of these parameters are read from a file external to the APM program, while the others are contained in DATA statements. Those parameters residing in the external file (FOR25) are read in the main program and set in COMMON blocks common to GNDCAL. The DATA statements are located in GNDCAL.

Only the calculations for the groundside model are performed by GNDCAL. Printing is done in a separate routine, GNDOUT.

The programming flow of GNDCAL is straightfoward. After initialization of variables is done, hourly enplanements and deplanements are calculated. They are then used to calculate the hourly terminal facility requirements for the airport: main lobby area, main lobby seats, ticket counter frontage, ticket counter area, baggage claim area, and the length of the baggage claim belt.

Access facility requirements (parking, curb and access/egress requirements) are calculated next. Again, hourly enplanements and deplanements are driving functions for the equations. The outputs of importance are: hourly short-term parking slot requirements; long-term parking slot requirements; the number of employees required; hourly employee parking slot requirements; long-term parking space requirements over and above spaces taken at the end of the day; net increase in long-term parking spaces at the end of the day; hourly enplanement and deplanement curb space requirements; and hourly requirements for inbound and outbound (access and egress) lanes of highway. Inbound and outbound lanes of highway are rounded up to the nearest integer. Peak values are calculated from the hourly requirements.

Upon completion of the calculations, GNDOUT is called and the results are printed and control is returned to the main program. This process is followed for daily runs. If annualization runs are being processed, GNDCAL compares the results of the calculation just computed for that day with the previously selected maximum calculations of preceding days. New maximum values are obtained if they exist. Upon completion of the calculations for the final day in an annualization run, GNDOUT is called for printing. This printing reflects only the maximum values ascertained from all days processed.

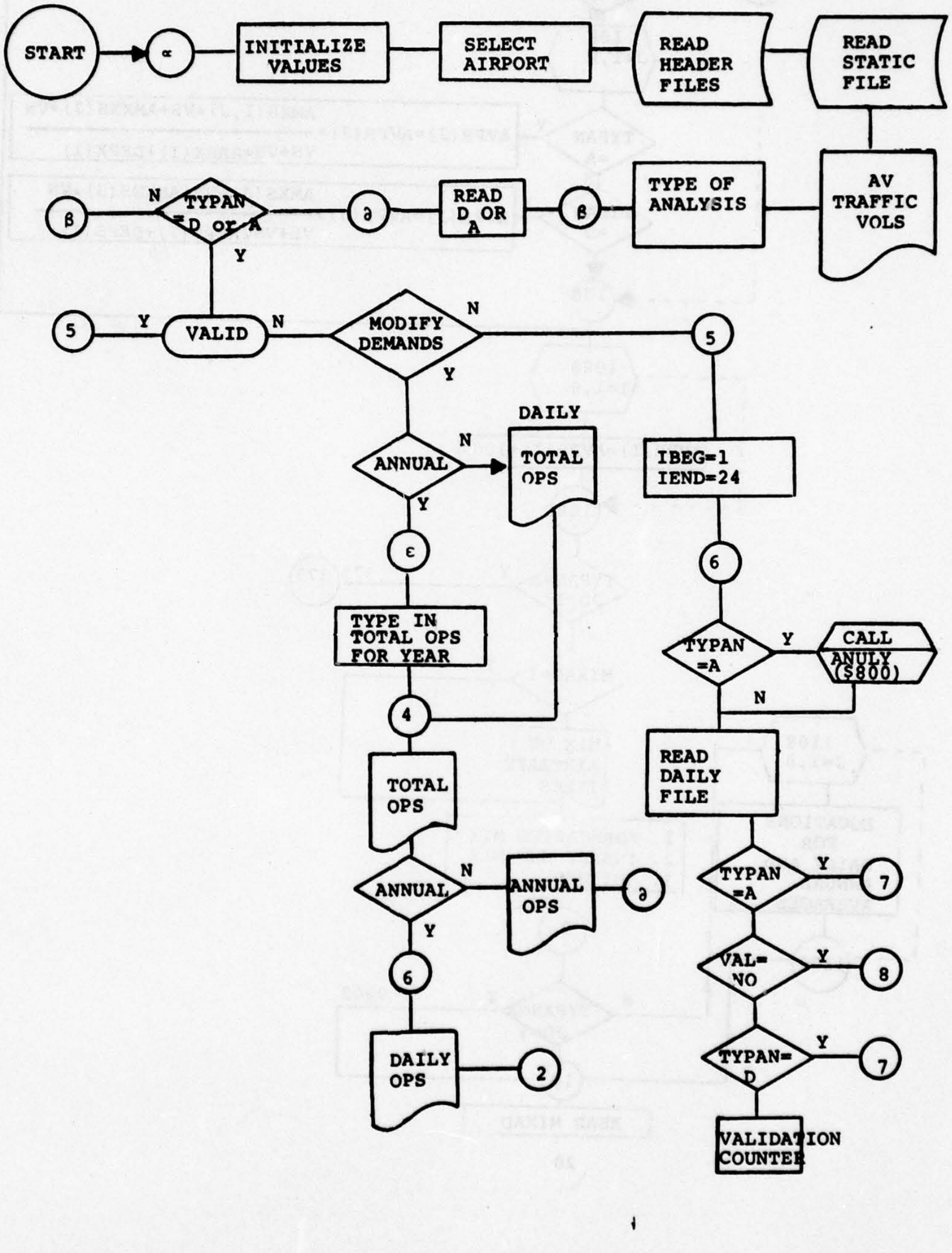
6.2.2 GNDOUT

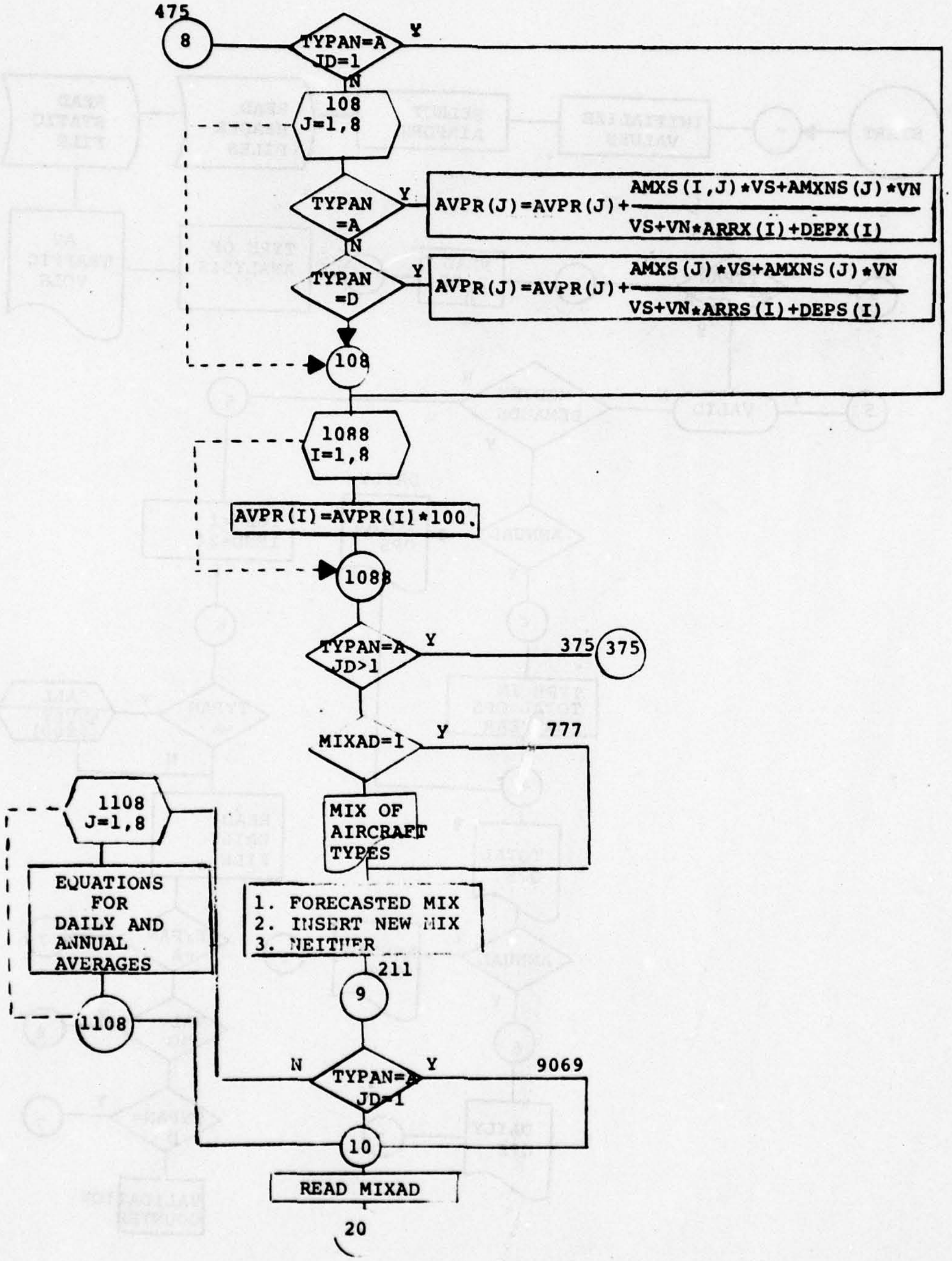
GNDOUT performs all printing for the groundside model. It is called by GNDCAL after each day is processed (daily runs only) or at the end of the processing of the last day (annualization runs).

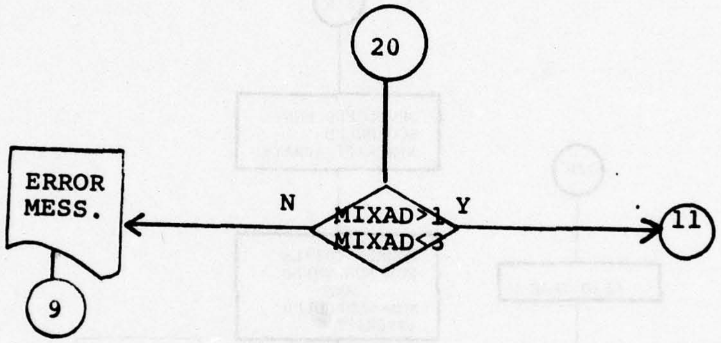
The printout appears in two forms: hourly requirements, and the peaks of the hourly requirements. The peaks reflect the maximum value of the output over a span of 24 hours, while the hourly requirements are the full compliment of output over 24 hours.

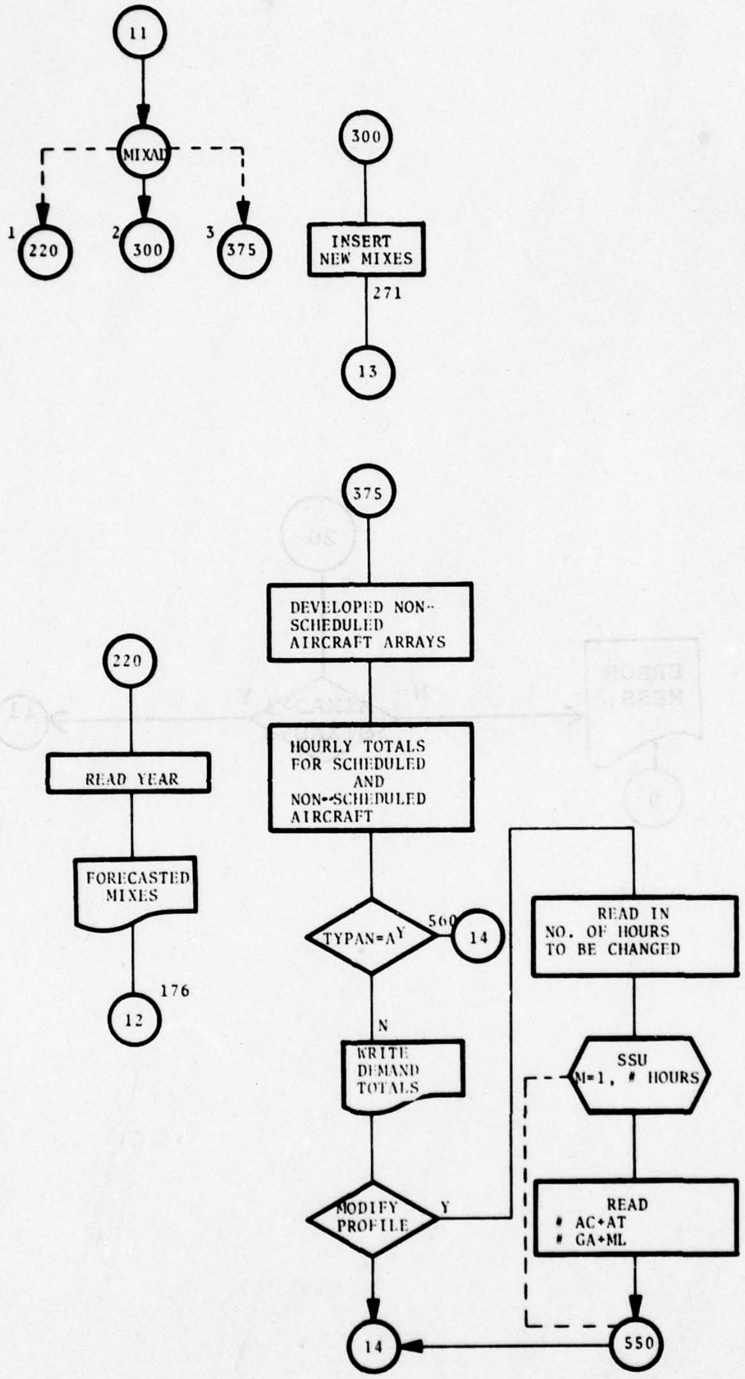
The hourly requirements are divided into three categories: terminal facility requirements, access facility requirements, and passenger movements. Each division is optional and controlled through the dialogue between the user and the program.

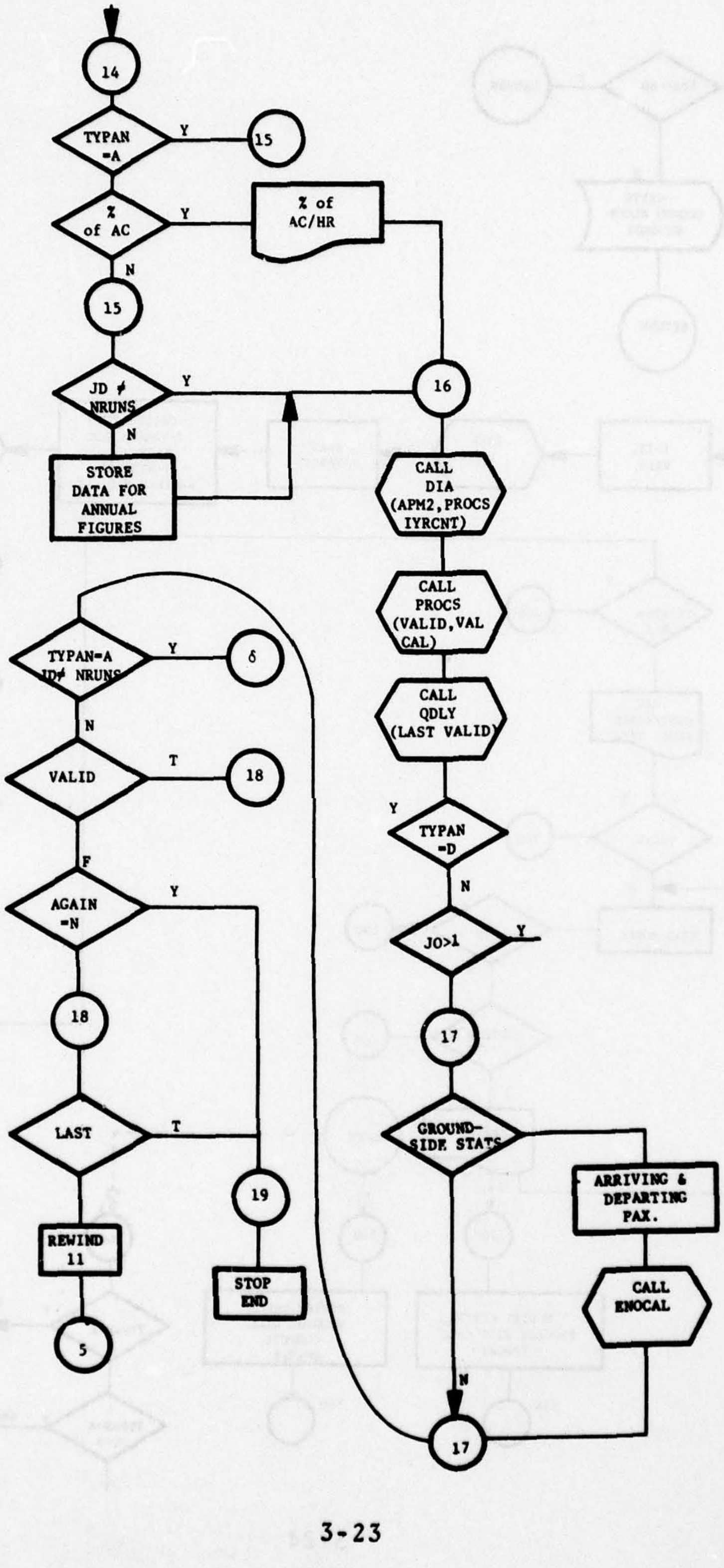
6.3 APM Program Flow Charts

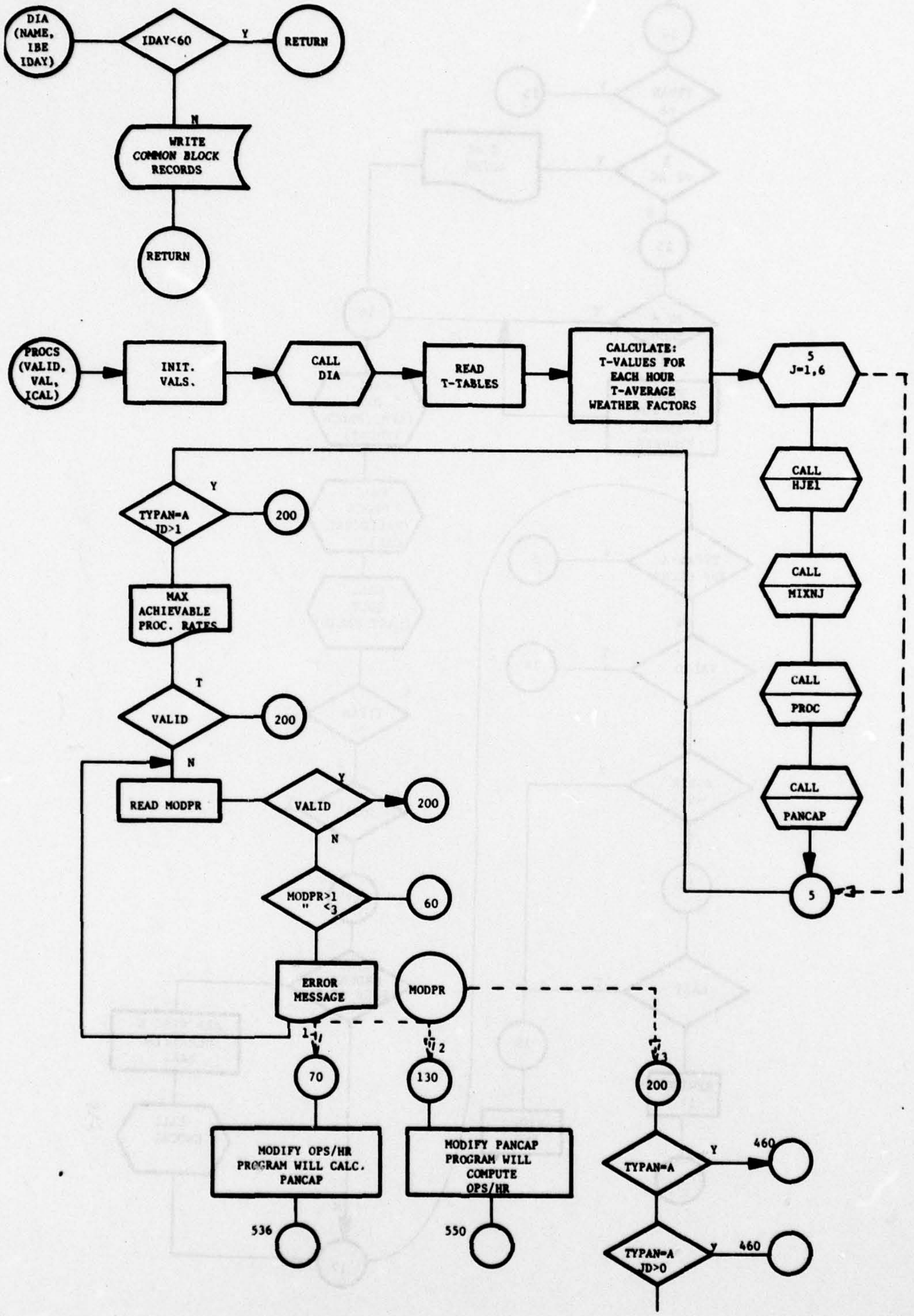


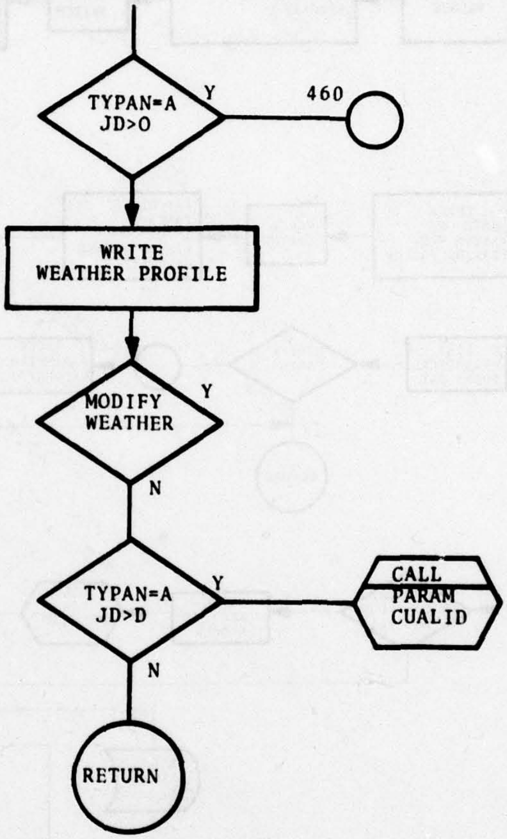

475
8
TYPAN=A
JD=1
Y
N
108
J=1,8
TYPAN
=A
Y
AVPR(J)=AVPR(J)+ AMXS(I,J)*VS+AMXNS(J)*VN / VS+VN*ARRX(I)+DEPX(I)
N
TYPAN
=D
Y
AVPR(J)=AVPR(J)+ AMXS(J)*VS+AMXNS(J)*VN / VS+VN*ARRS(I)+DEPS(I)
108
1088
I=1,8
AVPR(I)=AVPR(I)*100.
1088
TYPAN=A
JD>1
Y
375
375
MIXAD=1
Y
777
MIX OF
AIRCRAFT
TYPES
1. FORECASTED MIX
2. INSERT NEW MIX
3. NEITHER
211
9
1108
J=1,8
EQUATIONS
FOR
DAILY AND
ANNUAL
AVERAGES
1108
N
TYPAN=A
JD=1
Y
9069
10
READ MIXAD
20

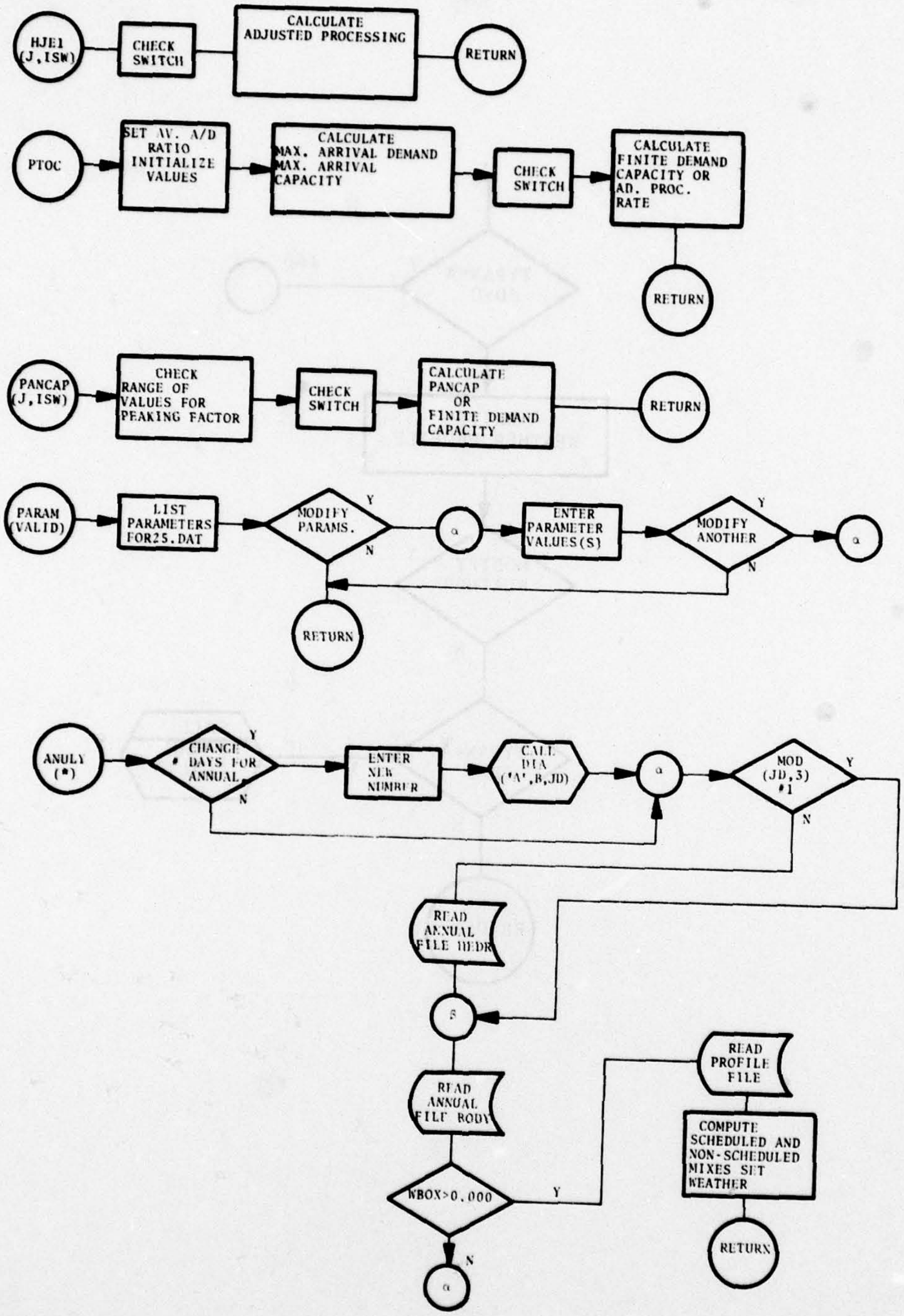

20
ERROR MESS.
N
MIXAD>1
MIXAD<3
Y
11
9

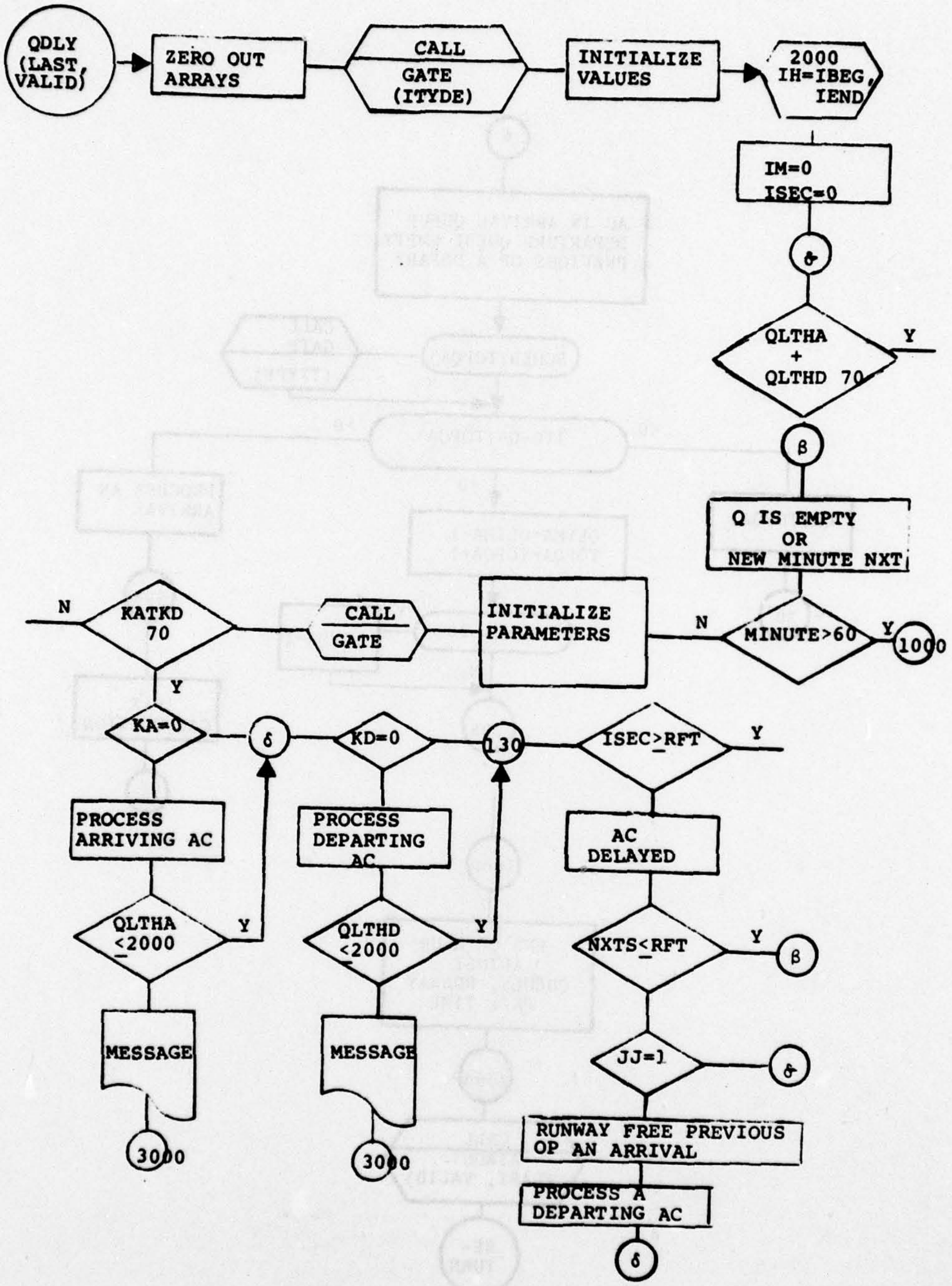

11
MIXAD
1
220
2
300
3
375
300
INSERT NEW MIXES
271
13
375
DEVELOPED NON-SCHEDULED AIRCRAFT ARRAYS
220
READ YEAR
FORECASTED MIXES
176
12
HOURLY TOTALS FOR SCHEDULED AND NON-SCHEDULED AIRCRAFT
TYPAN=A
Y
560
14
N
WRITE DEMAND TOTALS
MODIFY PROFILE
Y
14
READ IN NO. OF HOURS TO BE CHANGED
SSU M=1, # HOURS
READ # AC+AT # GA+ML
550

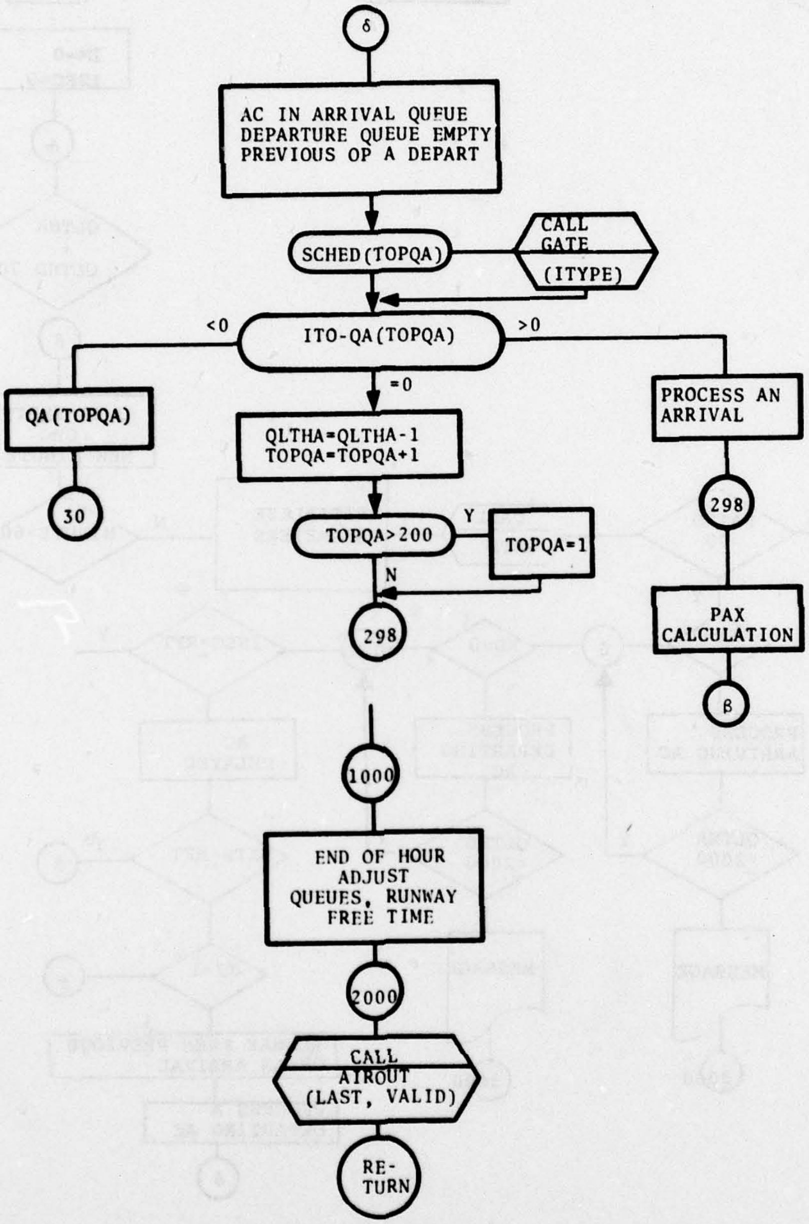

14
TYPAN =A
Y
15
% of AC
Y
% of AC/HR
N
15
JD ≠ NRUNS
Y
16
N
STORE DATA FOR ANNUAL FIGURES
CALL DIA (APM2,PROCS IYRCNT)
CALL PROCS (VALID,VAL CAL)
CALL QDLY (LAST VALID)
Y
TYPAN =D
N
JO>1
Y
17
GROUND-SIDE STATS
ARRIVING & DEPARTING PAX.
CALL ENOCAL
N
17
TYPAN=A JD≠ NRUNS
Y
δ
N
VALID
T
18
F
AGAIN =N
Y
18
LAST
T
19
REWIND 11
STOP END
5

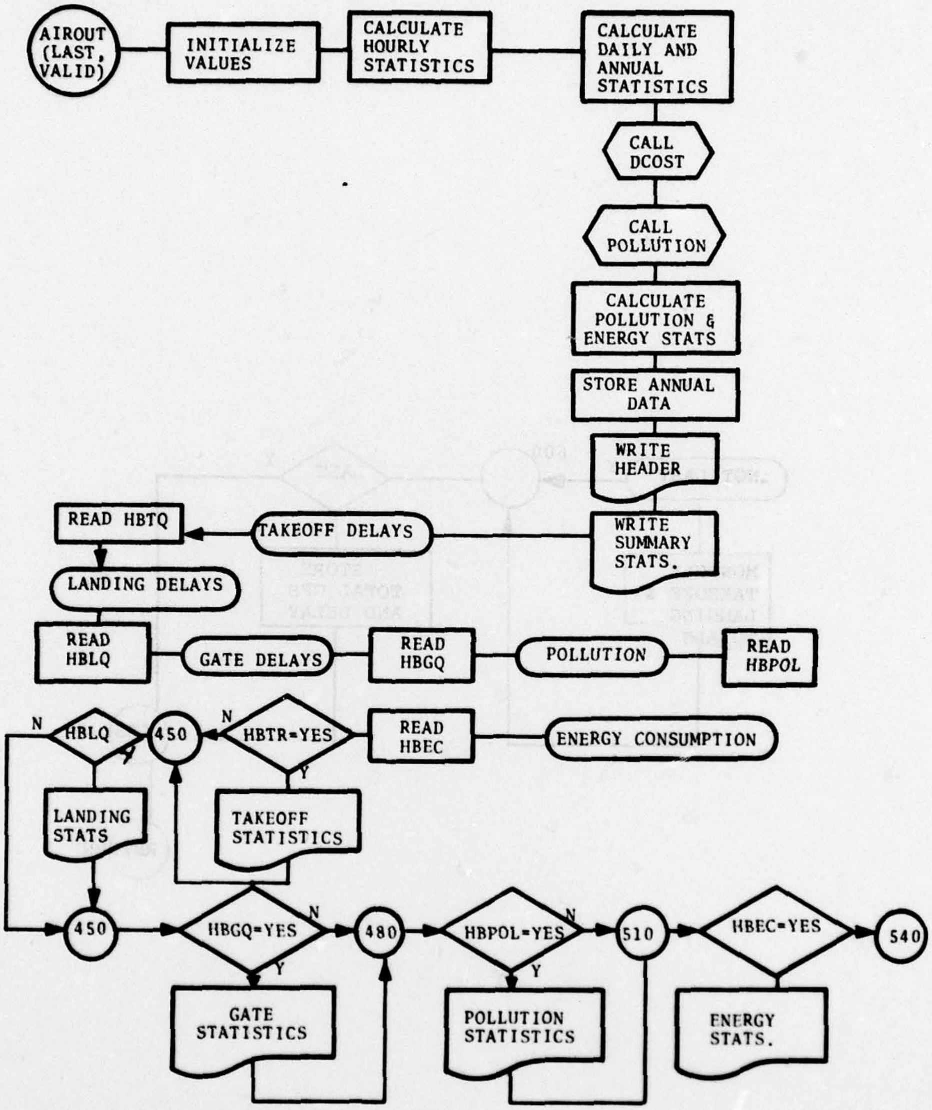

DIA (NAME, IBE IDAY)
IDAY<60
Y
RETURN
N
WRITE COMMON BLOCK RECORDS
RETURN
PROCS (VALID, VAL, ICAL)
INIT. VALS.
CALL DIA
READ T-TABLES
CALCULATE: T-VALUES FOR EACH HOUR T-AVERAGE WEATHER FACTORS
5 J=1,6
CALL HJE1
CALL MIXNJ
CALL PROC
CALL PANCAP
5
Y
TYPAN=A JD>1
200
MAX ACHIEVABLE PROC. RATES
T
VALID
200
N
READ MODPR
Y
VALID
200
N
MODPR>1 " <3
60
ERROR MESSAGE
MODPR
1
2
3
70
130
200
MODIFY OPS/HR PROGRAM WILL CALC. PANCAP
MODIFY PANCAP PROGRAM WILL COMPUTE OPS/HR
TYPAN=A
Y
460
TYPAN=A JD>0
Y
460
536
550

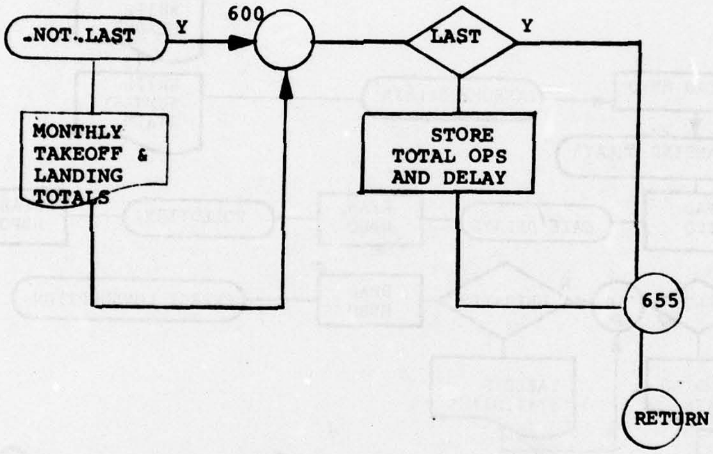

TYPAN=A
JD>O
Y
460
WRITE
WEATHER PROFILE
MODIFY
WEATHER
Y
N
TYPAN=A
JD>D
Y
CALL
PARAM
CUALID
N
RETURN

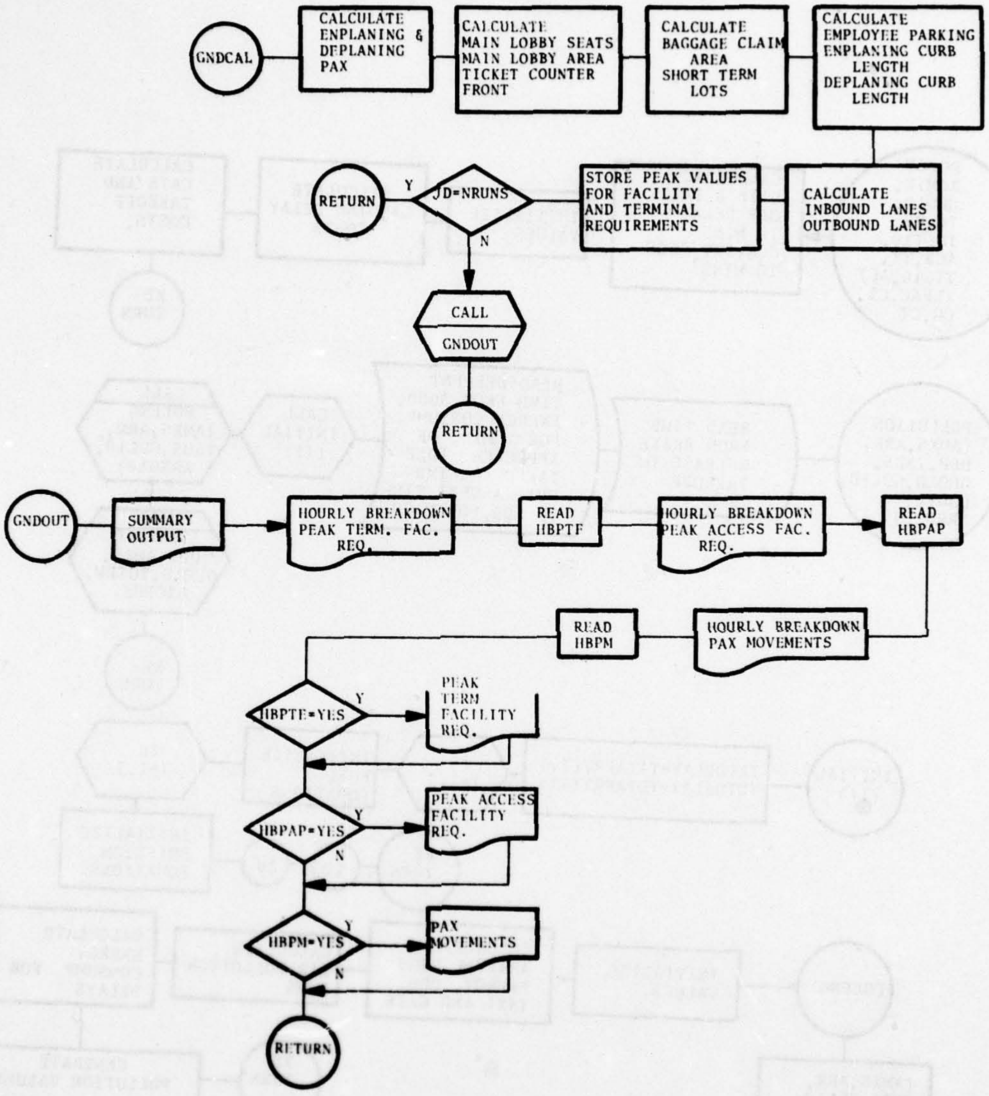

HJE1 (J,ISW)
CHECK SWITCH
CALCULATE ADJUSTED PROCESSING
RETURN
PTOC
SET AV. A/D RATIO INITIALIZE VALUES
CALCULATE MAX. ARRIVAL DEMAND MAX. ARRIVAL CAPACITY
CHECK SWITCH
CALCULATE FINITE DEMAND CAPACITY OR AD. PROC. RATE
RETURN
PANCAP (J,ISW)
CHECK RANGE OF VALUES FOR PEAKING FACTOR
CHECK SWITCH
CALCULATE PANCAP OR FINITE DEMAND CAPACITY
RETURN
PARAM (VALID)
LIST PARAMETERS FOR25.DAT
MODIFY PARAMS.
Y
N
α
ENTER PARAMETER VALUES(S)
MODIFY ANOTHER
Y
N
α
RETURN
ANULY (*)
CHANGE # DAYS FOR ANNUAL
Y
N
ENTER NEW NUMBER
CALL DIA ('A',B,JD)
α
MOD (JD,3) #1
Y
N
READ ANNUAL FILE HEDR
β
READ ANNUAL FILE BODY
WBOX>0.000
Y
N
α
READ PROFILE FILE
COMPUTE SCHEDULED AND NON-SCHEDULED MIXES SET WEATHER
RETURN

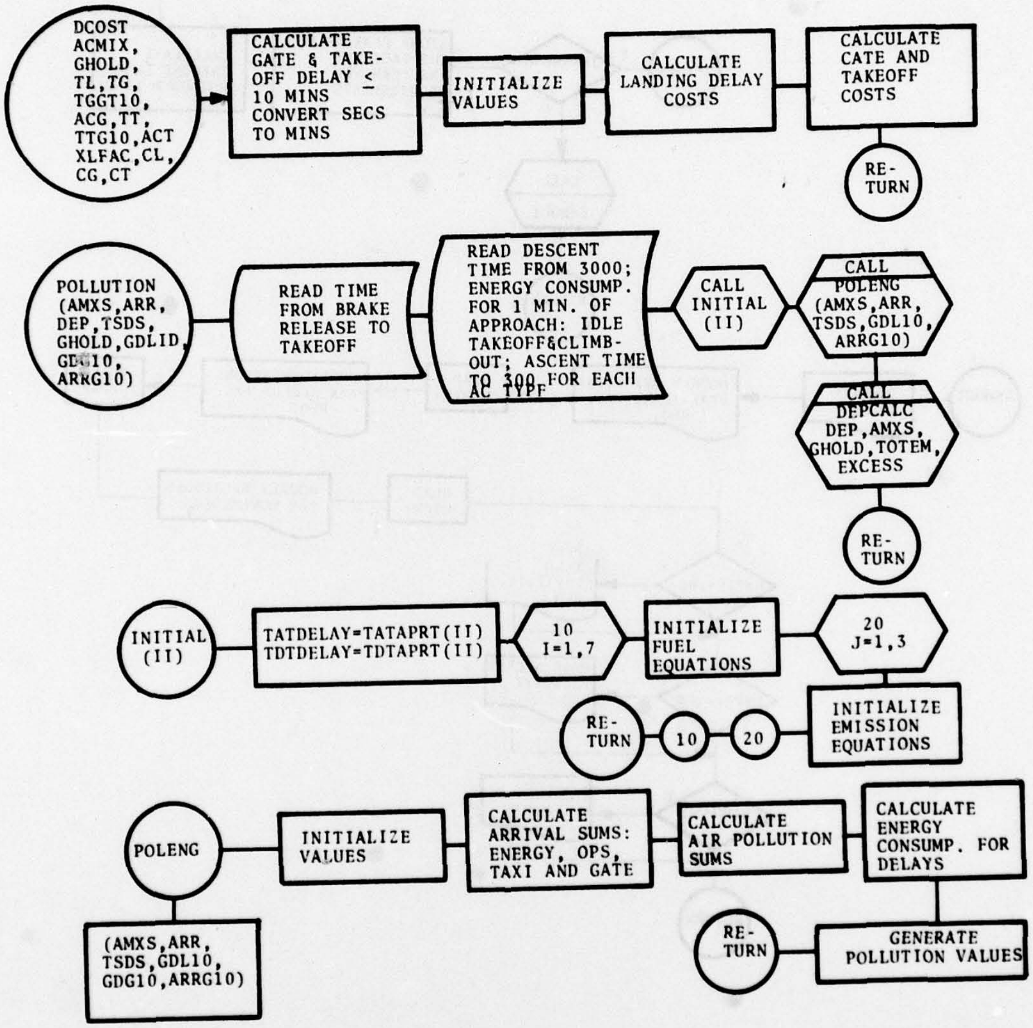

QDLY (LAST, VALID)
ZERO OUT ARRAYS
CALL GATE (ITYDE)
INITIALIZE VALUES
2000 IH=IBEG, IEND
IM=0 ISEC=0
α
QLTHA + QLTHD 70
Y
β
Q IS EMPTY OR NEW MINUTE NXT
MINUTE>60
N
Y
1000
INITIALIZE PARAMETERS
CALL GATE
KATKD 70
N
Y
KA=0
δ
KD=0
130
ISEC>RFT
Y
PROCESS ARRIVING AC
PROCESS DEPARTING AC
AC DELAYED
QLTHA <2000
Y
QLTHD <2000
Y
NXTS<RFT
Y
β
MESSAGE
MESSAGE
JJ=1
α
3000
3000
RUNWAY FREE PREVIOUS OP AN ARRIVAL
PROCESS A DEPARTING AC
δ

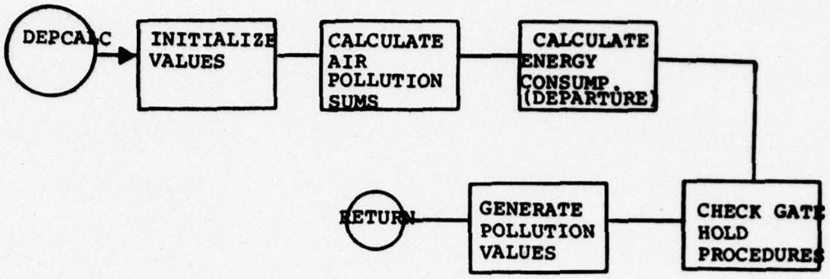

δ
AC IN ARRIVAL QUEUE
DEPARTURE QUEUE EMPTY
PREVIOUS OP A DEPART
SCHED(TOPQA)
CALL
GATE
(ITYPE)
<0
ITO-QA(TOPQA)
>0
=0
QA(TOPQA)
30
QLTHA=QLTHA-1
TOPQA=TOPQA+1
TOPQA>200
Y
TOPQA=1
N
298
PROCESS AN
ARRIVAL
298
PAX
CALCULATION
β
1000
END OF HOUR
ADJUST
QUEUES, RUNWAY
FREE TIME
2000
CALL
AIROUT
(LAST, VALID)
RE-
TURN

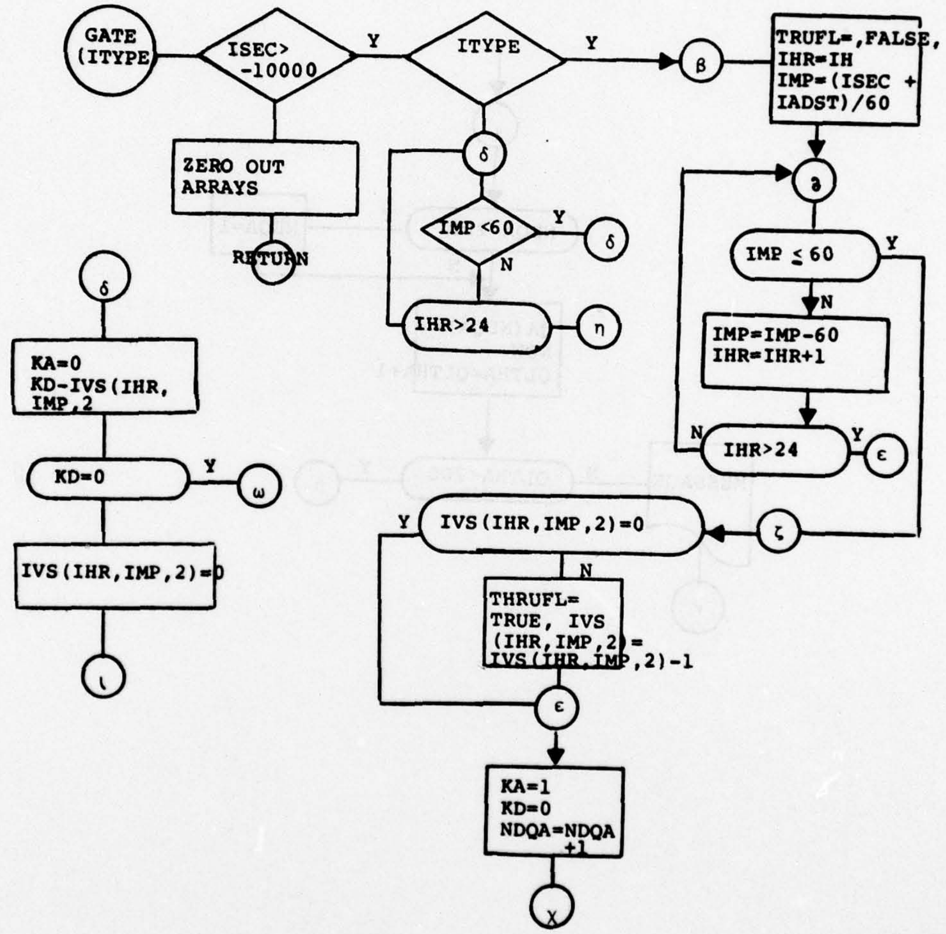

AIROUT (LAST, VALID)
INITIALIZE VALUES
CALCULATE HOURLY STATISTICS
CALCULATE DAILY AND ANNUAL STATISTICS
CALL DCOST
CALL POLLUTION
CALCULATE POLLUTION & ENERGY STATS
STORE ANNUAL DATA
WRITE HEADER
WRITE SUMMARY STATS.
TAKEOFF DELAYS
READ HBTQ
LANDING DELAYS
READ HBLQ
GATE DELAYS
READ HBGQ
POLLUTION
READ HBPOL
N
HBLQ
450
N
HBTR=YES
Y
READ HBEC
ENERGY CONSUMPTION
LANDING STATS
TAKEOFF STATISTICS
450
HBGQ=YES
N
Y
480
HBPOL=YES
N
Y
510
HBEC=YES
540
GATE STATISTICS
POLLUTION STATISTICS
ENERGY STATS.

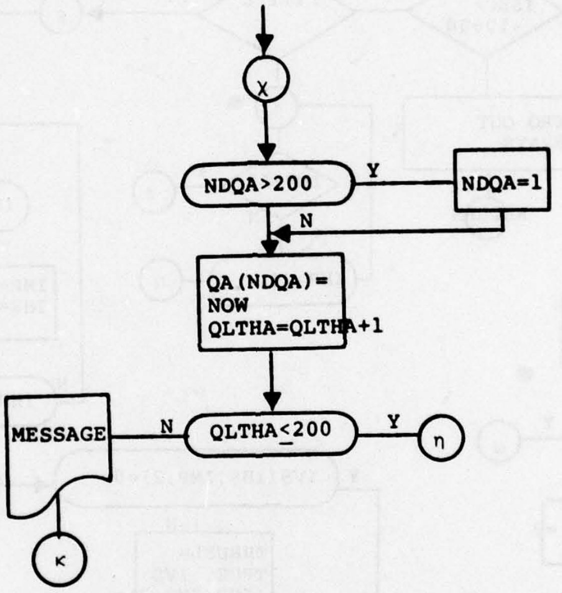

.NOT..LAST
Y
600
LAST
Y
MONTHLY TAKEOFF & LANDING TOTALS
STORE TOTAL OPS AND DELAY
655
RETURN

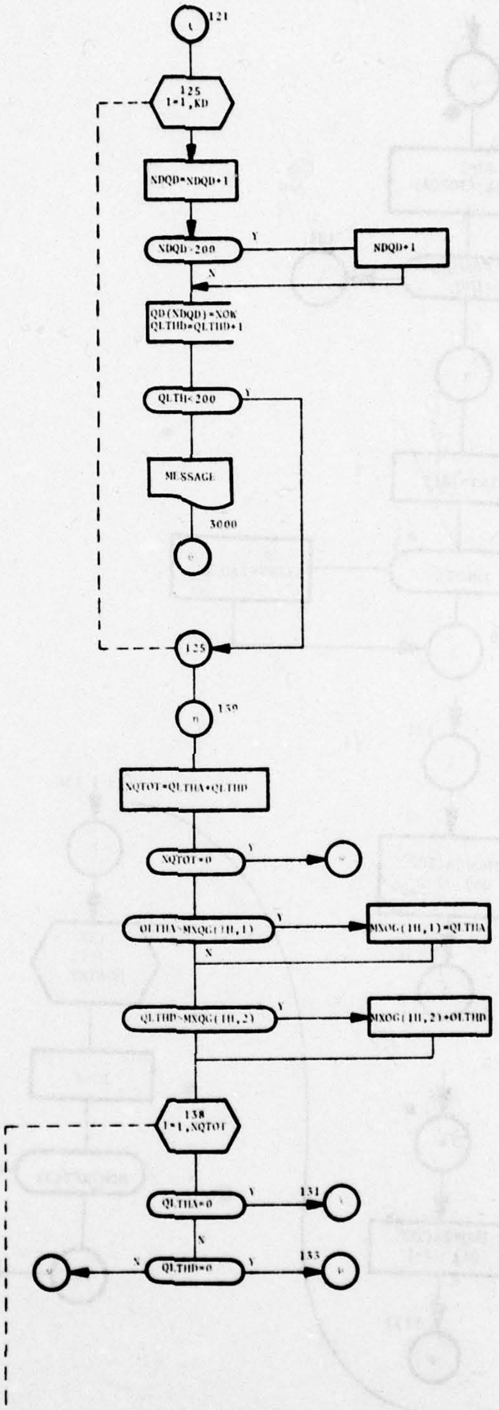

GNDCAL
CALCULATE ENPLANING & DEPLANING PAX
CALCULATE MAIN LOBBY SEATS MAIN LOBBY AREA TICKET COUNTER FRONT
CALCULATE BAGGAGE CLAIM AREA SHORT TERM LOTS
CALCULATE EMPLOYEE PARKING ENPLANING CURB LENGTH DEPLANING CURB LENGTH
CALCULATE INBOUND LANES OUTBOUND LANES
STORE PEAK VALUES FOR FACILITY AND TERMINAL REQUIREMENTS
JD=NRUNS
Y
N
RETURN
CALL
GNDOUT
RETURN
GNDOUT
SUMMARY OUTPUT
HOURLY BREAKDOWN PEAK TERM. FAC. REQ.
READ HBPTF
HOURLY BREAKDOWN PEAK ACCESS FAC. REQ.
READ HBPAP
HOURLY BREAKDOWN PAX MOVEMENTS
READ HBPM
HBPTF=YES
Y
PEAK TERM FACILITY REQ.
HBPAP=YES
Y
N
PEAK ACCESS FACILITY REQ.
HBPM=YES
Y
N
PAX MOVEMENTS
RETURN

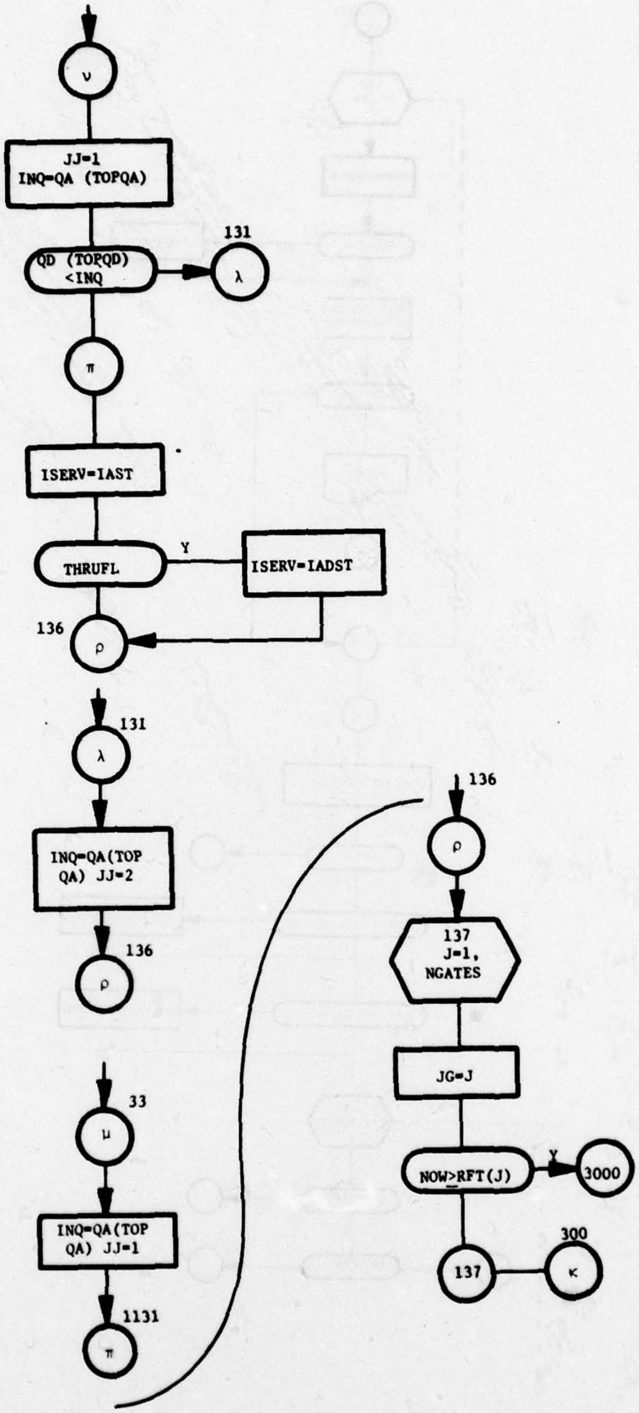

DCOST ACMIX, GHOLD, TL,TG, TGGT10, ACG,TT, TTG10,ACT XLFAC,CL, CG,CT
CALCULATE GATE & TAKE-OFF DELAY ≤ 10 MINS CONVERT SECS TO MINS
INITIALIZE VALUES
CALCULATE LANDING DELAY COSTS
CALCULATE CATE AND TAKEOFF COSTS
RE-TURN
POLLUTION (AMXS,ARR, DEP,TSDS, GHOLD,GDLID, GDG10, ARRG10)
READ TIME FROM BRAKE RELEASE TO TAKEOFF
READ DESCENT TIME FROM 3000; ENERGY CONSUMP. FOR 1 MIN. OF APPROACH: IDLE TAKEOFF&CLIMB-OUT; ASCENT TIME TO 300 FOR EACH AC TYPE
CALL INITIAL (II)
CALL POLENG (AMXS,ARR, TSDS,GDL10, ARRG10)
CALL DEPCALC DEP,AMXS, GHOLD,TOTEM, EXCESS
RE-TURN
INITIAL (II)
TATDELAY=TATAPRT(II) TDTDELAY=TDTAPRT(II)
10 I=1,7
INITIALIZE FUEL EQUATIONS
20 J=1,3
INITIALIZE EMISSION EQUATIONS
20
10
RE-TURN
POLENG
(AMXS,ARR, TSDS,GDL10, GDG10,ARRG10)
INITIALIZE VALUES
CALCULATE ARRIVAL SUMS: ENERGY, OPS, TAXI AND GATE
CALCULATE AIR POLLUTION SUMS
CALCULATE ENERGY CONSUMP. FOR DELAYS
GENERATE POLLUTION VALUES
RE-TURN

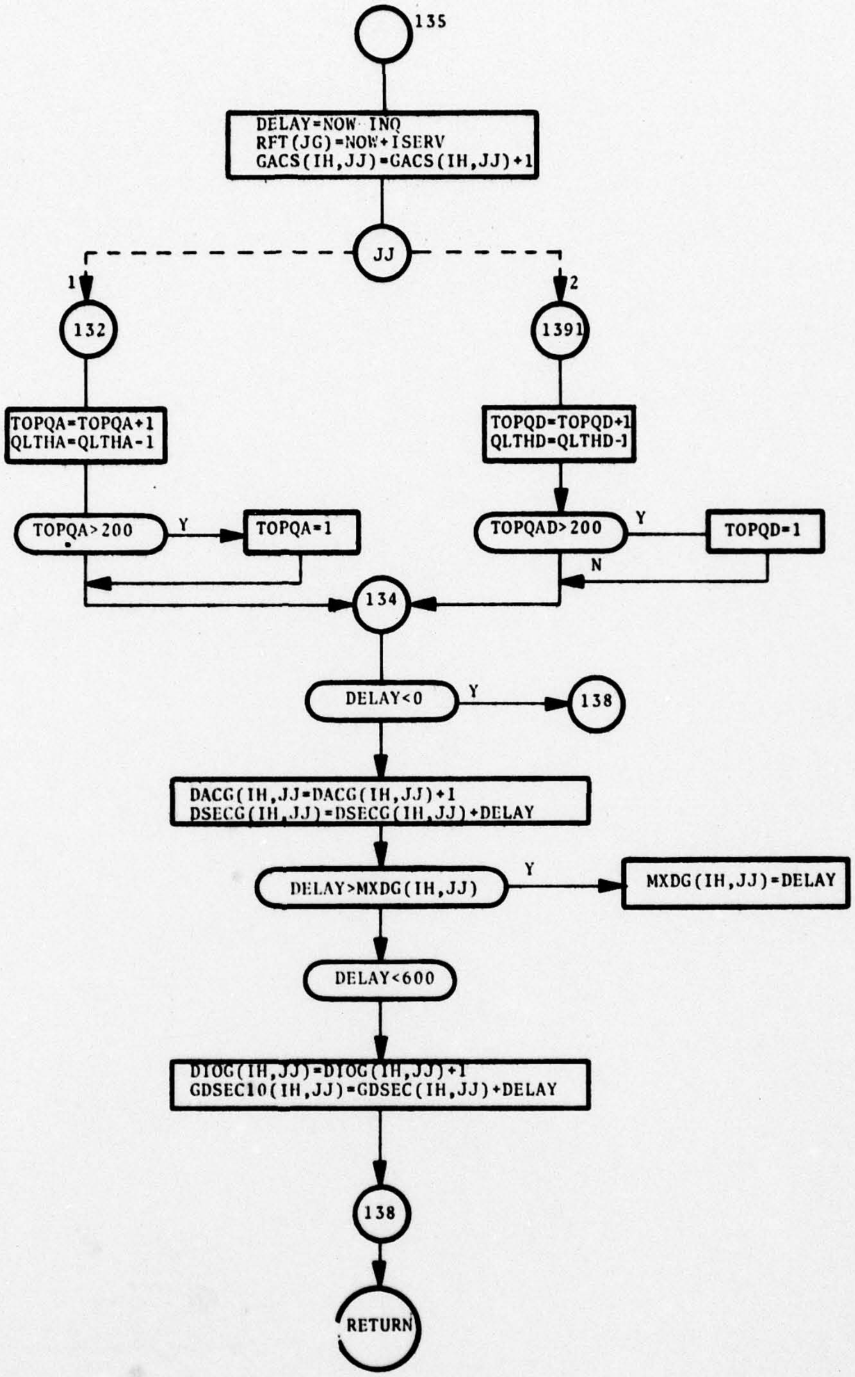

DEPCALC
INITIALIZE VALUES
CALCULATE AIR POLLUTION SUMS
CALCULATE ENERGY CONSUMP. (DEPARTURE)
CHECK GATE HOLD PROCEDURES
GENERATE POLLUTION VALUES
RETURN


GATE (ITYPE
ISEC> -10000
Y
ITYPE
Y
β
TRUFL=,FALSE, IHR=IH IMP=(ISEC + IADST)/60
ZERO OUT ARRAYS
RETURN
δ
IMP <60
Y
δ
N
IHR>24
η
∂
IMP ≤ 60
Y
N
IMP=IMP-60 IHR=IHR+1
N
IHR>24
Y
ε
δ
KA=0 KD-IVS(IHR, IMP,2
KD=0
Y
ω
IVS(IHR,IMP,2)=0
ι
Y
IVS(IHR,IMP,2)=0
ζ
N
THRUFL= TRUE, IVS (IHR,IMP,2)= IVS(IHR,IMP,2)-1
ε
KA=1 KD=0 NDQA=NDQA +1
χ


X
NDQA>200
Y
NDQA=1
N
QA(NDQA)=
NOW
QLTHA=QLTHA+1
MESSAGE
N
QLTHA<200
Y
η
κ


121
125
I=1,KD
NDQD=NDQD+1
NDQD>200
Y
NDQD=1
N
QD(NDQD)=NOW
QLTHD=QLTHD+1
QLTH<200
Y
MESSAGE
3000
125
139
NQTOT=QLTHA+QLTHD
NQTOT=0
Y
QLTHA>MXQG(IH,1)
Y
MXQG(IH,1)=QLTHA
N
QLTHD>MXQG(IH,2)
Y
MXQG(IH,2)=QLTHD
138
I=1,NQTOT
QLTHA=0
Y
131
N
QLTHD=0
N
Y
133


ν
JJ=1
INQ=QA (TOPQA)
QD (TOPQD)
<INQ
131
λ
π
ISERV=IAST
THRUFL
Y
ISERV=IADST
136
ρ
131
λ
INQ=QA(TOP
QA) JJ=2
136
ρ
33
μ
INQ=QA(TOP
QA) JJ=1
1131
π
136
ρ
137
J=1,
NGATES
JG=J
NOW>RFT(J)
Y
3000
137
300
κ


135
DELAY=NOW INQ
RFT(JG)=NOW+ISERV
GACS(IH,JJ)=GACS(IH,JJ)+1
JJ
1
2
132
1391
TOPQA=TOPQA+1
QLTHA=QLTHA-1
TOPQD=TOPQD+1
QLTHD=QLTHD-1
TOPQA>200
Y
TOPQA=1
TOPQAD>200
Y
TOPQD=1
N
134
DELAY<0
Y
138
DACG(IH,JJ=DACG(IH,JJ)+1
DSECG(IH,JJ)=DSECG(IH,JJ)+DELAY
DELAY>MXDG(IH,JJ)
Y
MXDG(IH,JJ)=DELAY
DELAY<600
DIOG(IH,JJ)=DIOG(IH,JJ)+1
GDSEC10(IH,JJ)=GDSEC(IH,JJ)+DELAY
138
RETURN

185 Copies